NASA TECHNICAL NOTE

NASA TN D-3624

NASA TN D-3624

c. 1

EVALUATION OF A NEW METHOD OF INTEGRATING THE ORBITAL EQUATIONS OF MOTION FOR USE IN SPACE NAVIGATION

by John D. McLean

Ames Research Center
Moffett Field, Calif.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • SEPTEMBER 1966

NASA TN D-3624

EVALUATION OF A NEW METHOD OF INTEGRATING THE ORBITAL EQUATIONS OF MOTION FOR USE IN SPACE NAVIGATION

By John D. McLean

Ames Research Center
Moffett Field, Calif.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – Price $2.50

TABLE OF CONTENTS

EVALUATION OF A NEW METHOD OF INTEGRATING THE ORBITAL EQUATIONS OF MOTION FOR USE IN SPACE NAVIGATION

By John D. McLean
Ames Research Center

SUMMARY

The applicability of a new method of integrating the orbital equations of motion, developed by Dr. J. M. A. Danby of Yale University Observatory, to the problem of space navigation is investigated. The investigation is carried out by means of a digital computer program written for the method. The transearth and translunar phases of a circumlunar trajectory are taken as sample problems, but the method can also be applied to any orbital mission. The gravitational effects of the sun, moon, earth, and earth's oblateness are considered.

A detailed description of the computer program, including listings, is presented. Data show the effects of various error sources and the trade off between computing speed and accuracy. The method is compared with Cowell's method on the basis of computer storage, integration time, and accuracy.

INTRODUCTION

In recent years considerable study has been devoted to the problems of navigation and guidance for manned space missions. Except for near-earth satellites, the astronaut will probably be provided with the capability, at least as a secondary system, for carrying out these tasks without the aid of ground-based equipment. If such an on-board navigation and guidance system is not to be restricted to limited emergency procedures, a digital computer will be required aboard the spacecraft.

One of the more demanding functions of such a computer is to solve the vehicle's equations of motion. The most commonly proposed method is the numerical integration of the differential equations of motion, although there are other possibilities such as the interpolation of stored data. Two well-known methods of setting up the equations to be integrated are Cowell's method in which the total accelerations acting on the spacecraft are integrated, and Encke's method in which differential equations for perturbations from an osculating conic are solved. Both of these methods require a complex integration routine, and any alternative which will substantially reduce the time and storage requirements is desirable. For on-board use the reduction in storage requirements is particularly desirable because the decrease in the number of components enhances reliability and reduces weight and power requirements. In addition, the problem of accurately starting the numerical solution of a system of differential equations is quite complicated. Since the estimated trajectory for a navigation system must be restarted every time new observational data are obtained, it would be desirable to avoid or minimize this difficulty.

The purpose of this report is to present an evaluation, from the standpoint of application to on-board computation, of a new approach to the problem of integrating the equations of motion. This integration procedure was developed by Dr. J. M. A. Danby of Yale University Observatory and is described in detail in references 1 and 2. As in Encke's method, the perturbations of the true orbit from a reference conic are computed,but the method reduces the integration of the perturbing accelerations from the solution of a set of differential equations to simple quadratures. Although a well-known mathematical technique is used to obtain the perturbations, Danby was, to the author's knowledge, the first to recognize the advantage of applying this technique to orbital equations. This method of solving the equations of motion will be referred to in the remainder of the report as "Danby's method." It will be shown how Danby's method can be used to reduce the problems discussed in the preceding paragraph.

The original application of Danby's method, as described in references 1 and 2, fits the trajectory with a small number of conics, called mean orbits. These mean orbits closely approximate the true orbit, including perturbations, thereby allowing it to be studied using closed form equations. The advantages of this procedure for many theoretical studies are obvious, but the accurate mean orbits are unnecessary for the specialized application of space navigation. Since the calculation of the mean orbits requires each portion of the trajectory to be integrated several times, it is desirable to avoid such calculations. Likewise, the method,as described in reference 2,uses eccentric anomaly as the independent variable in order to avoid inverting Kepler's equation. Since space navigation requires frequent processing of data at accurately known times, it would be desirable to use time as the independent variable. In this study several modifications have been made to Danby's method,as presented in the references,in order to eliminate these two difficulties without sacrificing computer storage requirements and speed. These modifications are described in detail and the modified system is compared with Cowell's method[1] for translunar and transearth trajectories.

NOTATION

a	semimajor axis of conic
A_L	lower bound on A_Q
A_Q	measure of validity of Simpson's rule
A_R	upper bound on position perturbation
A_S	lower bound on position perturbation
A_U	upper bound on A_Q

[1] Throughout this report "Cowell's method" will refer to Cowell's method of setting up the orbital equations of motion for integration rather than Cowell's numerical integration method.

f_1, f_2, f_3, f_4	functions of increment in eccentric anomaly
$f, g, \dot{f}, \dot{g}$	parameters of closed form conic equations
h	increment in time
h_E	increment in eccentric anomaly
I	a 3×3 unit matrix
$\overline{R}$	position vector
R	$\lvert\overline{R}\rvert$
$\overline{r}$	vector of small deviation in position
r	$\lvert\overline{r}\rvert$
S_T	total number of integration steps
t	time
u	6×1 matrix of perturbing accelerations
U	gravitational potential
U_c	potential of central body to which conic is referred
$\overline{V}$	velocity vector (equivalent to $\dot{\overline{R}}$)
V	$\lvert\overline{V}\rvert$
$\overline{v}$	vector of small deviation in velocity
v	$\lvert\overline{v}\rvert$
X, Y, Z	Cartesian components of vehicle's position
x	6×1 matrix (state vector of position and velocity deviations from conic)
y	transformed forcing function
∇	gradient operator
Δt	increment in time
$\Delta R, \Delta V$	magnitude of terminal position and velocity deviations of Danby solution from Cowell solution
θ	increment in eccentric (or hyperbolic) anomaly

Φ	state transition matrix
$\varphi_1, \varphi_2, \varphi_3, \varphi_4$	submatrices of Φ

Superscripts

T	transpose of a matrix
$\cdot$	derivative with respect to time
$\_$	a vector or 3×1 matrix

Subscripts

i,n	integers
o	initial value, usually referring to time of last rectification

THE INTEGRATION METHOD

The basic principles of Danby's method as presented in reference 1 are summarized here for the convenience of the reader. A more general coverage of the mathematical principles involved can be found in reference 3 or in texts dealing with vector and matrix differential equations or multidimensional control theory.

The basis of the method is the solution of the orbital equations of motion in terms of a reference conic and a set of associated perturbations. The computation of the perturbations is accomplished by simple quadratures through the use of the state transition matrix, in contrast to Encke's method which requires the solution of a system of differential equations. The transition matrix, $\Phi(t, t_o)$, is the matrix of first partial derivatives of the components (all the trajectory calculations discussed in this report are carried out in Cartesian coordinates) of position and velocity at time t with respect to the same quantities at time t_o. It can be shown that if x is a 6×1 matrix, or state vector, of position and velocity deviations from the reference trajectory then

$$x(t) = \Phi(t, t_o)x(t_o) \qquad (1)$$

The use of this matrix in the integration is explained as follows:

The orbital equations of motion may be written as

$$\ddot{\overline{R}} + \ddot{\overline{r}} = \nabla U(\overline{R} + \overline{r}) + u(\overline{R} + \overline{r}, t) \qquad (2)$$

where U is the gravitational potential due to the central body of the reference conic, $\overline{R}$ is the position vector on the conic, $\overline{r}$ is the vector of position deviations of the true orbit from the conic and $\bar{u}$ is the vector of perturbing accelerations. If it is assumed that r is sufficiently small, the true orbit may be approximated in terms of linear (first order) perturbations from the reference conic. In this case

$$\ddot{\bar{r}} = F_1(\bar{R})\bar{r} + \bar{u}(\bar{R}, t) \tag{3}$$

where F_1 is the 3×3 matrix of first partial derivatives of ∇U with respect to the components of $\bar{R}$.

Since $\bar{R}$ is a function only of position on the reference conic, hence of time, equation (3) represents a system of three linear second-order differential equations which can be rewritten as a first-order system to give

$$\dot{x} = Fx + u \tag{4}$$

where

$$u = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \bar{u} \end{pmatrix}$$

and

$$F = \begin{pmatrix} 0 & I \\ F_1 & 0 \end{pmatrix}$$

It can be shown that the solution of the homogeneous part of equation (4) is given by equation (1) and that the complete solution is

$$x = \Phi(t, t_o)x_o + \Phi(t, t_o)\int_{t_o}^{t} \Phi^{-1}(\tau, t_o)u(\tau)\,d\tau \tag{5}$$

Since the reference trajectory is a conic, the elements of Φ can be computed in closed form.[2] Furthermore, u is a function only of the position vector on the conic and time; that is, it does not depend explicitly on the perturbations. Thus the integrand in equation (5) is a known function of time and a simple quadrature formula, such as Simpson's rule, can be used to solve for each component of x separately.

[2] Since Φ also satisfies the homogeneous part of equation (4); that is,

$$\dot{\Phi} = F\Phi$$

where $\Phi(t_o) = I$, Φ may be found by numerical integration. In this case the reference orbit need not be a conic and the effects of perturbing accelerations on the transition matrices may be included. This is the method used to obtain Φ in Cowell program B discussed later in the report.

In addition, the problems involved in starting or restarting a numerical integration of a differential equation are eliminated. The elimination of a complex restart procedure is particularly important in a navigation system because the integration is stopped and restarted frequently to allow processing of observed data and revision of the estimated trajectory.

As the integration proceeds, the perturbations will grow until r is so large that equation (3) is no longer valid. When this occurs it is necessary to obtain a new reference conic (a process called rectification) for which equation (3) is valid. It is customary in most integration schemes to rectify only after r becomes too large, in which case the integration must be restarted at a point where r is still below the desired bound. It is possible, however, to rectify any time before r exceeds its bound.

APPLICATION TO ON-BOARD COMPUTATION

The purpose of the present study is to evaluate the utility of Danby's method for use in on-board space navigation systems, and a digital computer program was written for that evaluation. The details of the program, which was designed to be incorporated into a complete (simulated) guidance and navigation system at a later date, are presented in the appendixes. Appendix A contains the formulas used for the reference conic and a derivation of the equations (different from the equations of refs. 1 and 2) for the transition matrices. The equations for the components of the forcing function and the quadrature formula are given in appendixes B and C, respectively, and the program listings are given in appendix D.

As was stated earlier, certain modifications were made in the mechanization in order to use Danby's method to best advantage in the space navigation problem. These modifications are discussed in detail in the following paragraphs.

The Reference Orbit and Rectification

The first modification is in the choice of the reference conic and this in turn requires a different rectification procedure. In the application described in references 1 and 2, the integration is started at time t_0 using the osculating conic as the reference orbit and rectification is carried out at some time, t_1, when r exceeds a predetermined value. At that time a new reference conic is computed which has a different initial velocity and passes through the position on the perturbed orbit at t_1. The integration and rectification are repeated several times, always starting at t_0 and going to progressively later values of t_1, until a reference conic, or mean orbit, is obtained which approximates a large segment of the true orbit very closely (in position). Because of this repeated integration, it is unnecessary to constrain r to very small values, and a large number of integration steps are made between rectifications.

Since the determination of an accurate mean orbit is not necessary for space navigation, it was decided to eliminate the repeated integration and use the osculating conic as the only reference orbit. In this case a more stringent restriction must be put on r in order to obtain the desired accuracy, and it was found that over most of the trajectory rectification must occur after each integration interval. If the value of r is used to determine when to rectify, the maximum allowable value must be exceeded before it is known whether rectification is necessary. Thus the new conic must originate at the beginning of the last integration step, and the integration over this last step must be repeated. If this rectification procedure is used with the osculating conic as the reference trajectory, most of the trajectory will be integrated twice. For this reason the program was arranged to require rectification after each integration step. There is negligible penalty in computing time or storage for such frequent rectification, but it was found, as will be discussed later, that excessive round-off errors are encountered near the centers of attraction. This difficulty can be avoided while still maintaining efficient use of computing time by the use of double precision for the computation of the reference conic. An alternate method would be to take several integration steps between rectifications near the centers of attraction while rectifying after every integration step elsewhere.

Choice of the Independent Variable

The second modification in the use of Danby's method was in the choice of the independent variable. Danby (ref. 2) recommended that the eccentric anomaly of the Keplerian orbit be used in order to avoid the iteration necessary for the inversion of Kepler's equation. While this choice has obvious advantages for a trajectory study, it is not particularly suitable for an on-board navigation system. Such a system requires that quantities from which the trajectory can be determined be observed at intervals. By some method a trajectory is estimated and the values that the observations would have if the spacecraft were on the estimated trajectory are computed. Because of observational errors the actual trajectory can never be known, and that estimated trajectory which minimizes the differences or residuals between the actual and computed observations is found. Since the times of the observations are fixed, they will occur at different eccentric anomalies for each estimated trajectory. Hence, the eccentric anomalies at the times of the observations are unknowns to be determined, while the times associated with them can be known quite accurately.

This difficulty presumably could be overcome by some sort of interpolation scheme, but the cost in computer storage would be large. In addition, if a Kalman filter type (ref. 4) of trajectory determination is used, the transition matrices between observations would be needed, and these must relate the states at the two different times. For these reasons, time was chosen as the independent variable. However, in order to minimize the number of solutions of Kepler's equation, the integration was set up as outlined below.

The Integration

Although the program in its final form was set up to rectify after each integration step, this discussion is presented so as to be equally applicable to the case where several integration steps are made between rectifications. (The program presented in appendix D can be modified fairly easily to take several steps between rectifications.) As an aid to clarity in discussion, t_o is defined as the time of the last rectification while θ is the change in eccentric anomaly between t_o and a subsequent time.

Assume that equation (5) has been integrated from t_o to t_n and it is desired to continue the integration to t_{n+2}. This integration proceeds as follows:

(1) At t_n, the elements of $u(t_n)$ and $\Phi(t_n,t_o)$ have been computed in the previous step unless $t_n = t_o$. In the latter case Φ is the unit matrix and $u(t_n)$ must be computed.

(2) The forcing function u is transformed to time t_o by

$$y_n = \Phi^{-1}(t_n, t_o)u(t_n)$$

where the transformed forcing function y is the integrand in equation (5).

(3) The angle θ is increased from θ_n to $(\theta_n + h_E)$ and the associated position and velocity are computed along with the time, t_{n+1}, the transition matrix $\Phi(t_{n+1}, t_o)$, and $u(t_{n+1})$. Then

$$y_{n+1} = \Phi^{-1}(t_{n+1}, t_o)u(t_{n+1})$$

(4) The angle θ is increased to $(\theta + 2h_E)$ and the conic is extended to t_{n+2}. If t_{n+2} is smaller than (or equal to) the next time at which an observation, velocity correction, or termination of the flight is to occur

$$y_{n+2} = \Phi^{-1}(t_o, t_{n+2})u(t_{n+2})$$

is computed. This amounts to making two equal increments in θ and accepting the associated change in time. If t_{n+2} is greater than the next desired time of stopping, the value of $(\theta_{n+2} - \theta_n)$ which will make the two increments equal is found by iteration. This change in eccentric anomaly is halved to give a new h_E and the process is repeated from (2). (In the remainder of the report the term "step size" will refer to h_E and "integration step" to the interval $t_n \leq t \leq t_{n+2}$.)

(5) The y_n, which constitute a set of values of the forcing function at times t_n, t_{n+1}, and t_{n+2} transformed to the rectification time, t_o, are integrated by quadratures.

These are sufficient data for the quadrature of each component of the forcing function by Simpson's rule to give

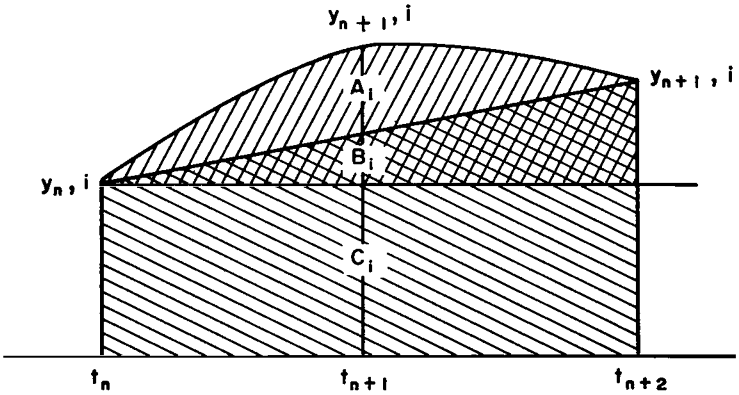
$$\int_{t_n}^{t_{n-2}} \Phi^{-1}(\tau, t_n)u(\tau)d\tau = a_O y_n + a_1 y_{n+1} + a_2 y_{n+2} \tag{6}$$

but the quadrature is complicated by the fact that the time increments are unequal. Simpson's rule is derived on the assumption that the curves to be integrated can be fit with a quadratic polynomial in the independent variable which, in this case, is time. If the time intervals are equal, the coefficients of the polynomial are constants, but for unequal intervals they must be calculated each time. Expressions for these coefficients are derived in appendix C.

Adjustment of the Step Size

The incremental eccentric anomaly h_E over which the quadratic approximation is valid changes markedly over the length of a lunar trajectory. This change is most radical near the centers of attraction and can be attributed mainly to the rapid change of the elements of the transition matrices in this region (see ref. 5). Thus, for greatest efficiency it is desirable to adjust the step size as the integration proceeds. If the eccentric anomaly were the independent variable, the fourth differences of the y_n could be computed and would give a good indication of the accuracy of the quadrature (see ref. 6). The step size could then be adjusted to be the maximum size which would restrict the magnitude of the fourth difference, and hence, of the error, to the desired limit.

It is possible to derive an expression which is equivalent to the fourth difference for unequal time increments, but the resulting computation is rather cumbersome. Since simplicity is one of the objects of this investigation, it was decided to try a simpler procedure which was found empirically to be quite satisfactory. This procedure is explained with the aid of sketch (a).



Sketch (a)

The area under the quadratic curve in the sketch represents the ith component of the integral given by equation (10). Since the perturbing accelerations are known to be smoothly varying functions of time, it was reasoned that the amount of curvature in the quadratic curve passing through the $y_{n,i}$ could be used as a measure of the size of higher order fluctuations. To assume $y_{n,i}$ to be a constant would be the simplest approximation and would result in an integral equal to the area C_i. A linear approximation would add the area B_i to this integral while the quadratic curve would add the area given by the algebraic sum $(A_i + B_i)$. A measure of the curvature of the quadratic curve is therefore the ratio of A_i to $(A_i + B_i)$. The six components of the integral were considered collectively as follows: Let r_Q^2 be the sum of the squares of the A_i representing position deviations while v_Q^2 is the sum of the squares of the A_i representing velocity deviations. Similarly, let r_T^2 and v_T^2 represent the sums of squares of $(A_i + B_i)$. Finally, define A_Q as

$$A_Q = \frac{r_Q^2}{r_T^2} + \frac{v_Q^2}{v_T^2}$$

The step size is automatically adjusted so that the A_Q remains in the range

$$A_L < A_Q < A_U$$

where A_L and A_U are input constants.

The step size must be small enough for the linearity assumption (that the true orbit can be expressed accurately in terms of small perturbations from the reference conic) to still be valid. This means that the error, $\delta\bar{r}$, in the position perturbation vector, $\bar{r}$, must be small compared to the position vector of the reference conic. If one uses the error analysis of reference 1 on a simple one-dimensional example, it is seen that over most of the trajectory δr will be roughly proportional to r^2/R_2, where R_2 is the distance from the vehicle to the perturbing body. However, when the distance, R_1, to the central body is small compared to R_2 (particularly when the earth is the central body), δr may become proportional to r^2/R_1. Since it is desired to bound $\delta r/R_1$, which is proportional either to r^2/R_2^2 or r^2/R_1R_2, it was decided for simplicity to require

$$A_S \leq \frac{r}{R} \leq A_R$$

where R is the smaller of R_1 and R_2, and A_S and A_R are input constants.[3]

The program in appendix D is arranged so that if either A_Q or r/R exceeds its upper bound after a given integration step, the step size is halved and the integration step is repeated. If both quantities drop below their lower bounds, the step size is doubled for the next interval. Part of the digital computer study discussed in the next section was devoted to the problem of choosing values for the parameters A_S, A_R, A_L, and A_U which control the value of h_E.

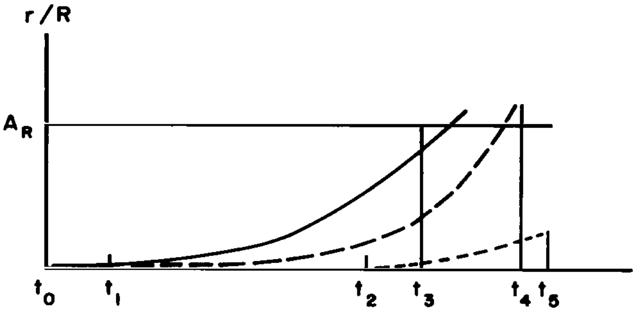
[3] If it is not desired to rectify every integration step, then a good criterion is to rectify when r/R exceeds A_R.

THE DIGITAL COMPUTER STUDY

This section of the report presents the results of a digital computer study used to evaluate the application of Danby's method outlined in the previous section. The first part of the discussion deals with the choice of desirable values for the parameters A_S, A_L, and A_U and some of the data used in establishing these values are used to illustrate the effects of round-off error. It was found that A_R must be chosen on the basis of a trade-off between speed and accuracy, and the second part of the discussion deals with this trade-off and with the time required to invert Kepler's equation. Finally, Danby's method is compared with Cowell's method in terms of the time and computer storage required to perform various tasks.

Choice of System Parameters

The first step in the numerical study was to determine desirable values for the upper and lower bounds, A_S, A_R, A_L, and A_U discussed earlier. These bounds establish a band of allowable error for each integration step, but the location of the actual error within this band is random. For example, the position deviation, $\bar{r}$, from the reference conic and the corresponding error in $\bar{r}$ increase nonlinearly with time and the rate of increase also varies along the trajectory. Sketch (b) illustrates how this can affect the accuracy of the integration. Suppose the nth integration step covers the time from t_o to t_3 so that the perturbation (shown in the sketch by the solid line) from the reference conic, and hence the error due to nonlinearity, nearly attains its maximum allowable value. Now consider a change in A_L. This might affect the previous part of the trajectory so that the nth integration step covers the time interval from t_1 to t_4 (dashed line) in which case r/R exceeds A_R, and h_E is reduced. With h_E reduced the nth integration step terminates at t_2 while the (n+1)th step (dotted curve) covers the time from t_2 to t_5. With the reduced value of h_E the perturbation and corresponding error due to nonlinearity remain much smaller than when the nth integration step begins at t_o. On the other hand, while reducing h_E reduces the error due to nonlinearity, it increases the possibility of round-off error. When the step size and corresponding perturbations are made quite small, then part of the perturbation is smaller than the least significant



Sketch (b)

figure in the corresponding component of the conic position or velocity and is lost as round-off error during rectification. (In fact, the perturbation could be made so small that the conic position and velocity would not be altered by rectification.)

In the interest of efficient use of computing time it would be desirable to make the width of the error band zero and set the upper bounds to give the maximum error consistent with the desired accuracy. If one attempts to approach this situation by the use of upper and lower bounds which are nearly equal, the program will back up an excessive number of times for step size reduction. On the other hand, if the upper and lower bounds are too far apart h_E will be kept unnecessarily small. Thus it was decided to choose the lower bound which would give the most efficient operation for each corresponding upper bound and to use the latter parameters to establish the accuracy of integration. The following paragraphs describe the method used in establishing A_S, A_L, and A_U and show the effect of round-off error on accuracy.

Choice of lower bounds.- To establish a reasonable value for A_S, a number of transearth trajectories were computed with the step size adjusted only by r/R. (Reasonably accurate results could be obtained by proper choice of the initial step size.) Figure 1 shows S_T, which is defined as the total number of integration steps including repeats for the reduction of h_E, plotted as a function of A_S/A_R for various values of A_R. The value of S_T is proportional to the integration time and thus is a measure of the efficiency of the program for a given value of A_R; S_T is not extremely sensitive to A_S/A_R for the larger values of A_R, but as that parameter is reduced, a definite minimum is obtained for a ratio of about 1/5. As a result of these data, A_S was constrained to be 1/5 of the value of A_R for the remainder of the study.

With A_S/A_R fixed, it is now possible to determine a similar relationship between A_L and A_U. In figure 2 S_T is plotted for the transearth trajectory as a function of A_L/A_U for various values of A_R and A_U. No optimum value of A_L/A_U can be seen from these data, but it is clear that the smaller ratios require an excessive number of steps. Likewise, most of the curves flatten out or rise slightly for the largest ratio. These results imply that A_L should be about 1/10 the value of A_U.

Choice of A_U.- Because of the randomness of the errors in integration discussed previously, one integration is not sufficient to establish the accuracy associated with a given set of values of A_R and A_U and it is desirable to obtain some estimate of the maximum likely error. In the trajectories used for figure 2, it was found that for given values of A_U and A_R, the greatest differences in terminal position and velocity usually occurred between trajectories having the maximum and minimum values of A_L. For this reason the only values of A_L/A_U used in determining A_U were 0.01 and 0.25. The values of A_R and A_U were the same as those for figure 2, except for one additional value (0.01) for A_U. This set of trajectories was integrated with the osculating conic computed in double precision because when single precision is used the influence of A_U is obscured by round-off errors. The X component ΔX (see appendix B for the definition of the coordinate system) of the terminal position deviation from the Cowell solution for these sets of trajectories is plotted in figure 3 as a function of A_U for different values of A_R. The

X component comprises most of the deviation for this particular trajectory and was used to preserve sign information. The results for the single-precision reference conic are also presented for use later in the discussion of round-off error.

Since the use of double precision essentially eliminates round-off errors, the errors in these solutions must be attributed to nonlinearity and inaccuracies arising from the use of Simpson's rule. For $A_R \leq 10^{-4}$ the deviations of the solutions for Danby's method (double precision) from the Cowell solution are always less than 0.5 km. The question of a "correct" solution when the deviations are so small is somewhat nebulous and neither method can be considered as more accurate than the other. A considerably more detailed study of both Cowell's and Danby's methods would be required to resolve the accuracy question further, and this study was not made since these results are satisfactory for navigation and guidance.

The lack of an absolutely correct answer does not prevent the use of these data in establishing the effects of various error sources and determining a reasonable value for A_U. For the two smallest values of A_R it is evident that reducing A_U from 0.2 to 0.1 improves the accuracy of the solution, but further reduction has negligible effects. The same trend is also seen in the curves for $A_R = 10^{-4}$, but it is obscured by increased spread between the solutions for maximum and minimum A_L that results from nonlinearity. On the basis of these results it was decided that the values of A_U should be set at 0.1 for best accuracy and efficiency.

Effects of round-off errors.- We now compare the results of the single precision computations (of the reference conic) with those of using double precision in order to assess the effects of round-off error. The double and single precision solutions corresponding to the same combinations of A_R, A_U, and A_L used identical sequences of h_E over the entire trajectory. For this reason the perturbations computed at each step for the two solutions are nearly the same, and the terminal deviation of the single precision solution from that for double precision must be attributed mainly to round-off.

The differences between these curves are within the accuracy limits of the Cowell integration for most of the trajectories, but a definite inaccuracy due to round-off is evident for all single precision solutions with $A_R = 2\times10^{-5}$ (fig. 3(a)). On the other hand, when A_R is increased to 2×10^{-4} (fig. 3(d)), the single precision results are more accurate than those for double precision. This indicates that for $A_R = 2\times10^{-4}$ the errors in the perturbations resulting from nonlinearity have become comparable to the least significant figure of the single precision conic position or velocity. In this case the part of the perturbation lost in round-off is entirely erroneous so that the round-off "errors" sometimes improve the accuracy.

Consequently, it appears that by the proper choice of A_U and A_R, it is possible to obtain good accuracy without resorting to double precision. However, in the vicinity of the centers of attraction, the constraint on A_Q reduces r/R well below the value of A_R because of the rapid change in the transition matrices mentioned earlier. The round-off errors caused by this reduction in step size are evident in the curves for figure 3 when $A_U = 0.01$.

This round-off error has no significant effect in the transearth case (if $A_U > 0.01$). For the translunar case, however, the round-off errors for $A_U = 0.1$ are large enough to cause sizable errors.

This phase of the digital computer study resulted in the choice of constant values of 0.2 for A_S/A_R, 0.1 for A_L/A_U, and 0.1 for A_U. The next data to be presented will show how, with these parameters fixed, it is possible to establish a trade-off between speed and accuracy by changing the value of A_R.

Accuracy and Integration Times

Using the values of A_S, A_L, and A_U determined in the previous section, it is now possible to consider the trade-off between integration time and accuracy. It should be pointed out that the integration times to be presented here were obtained from a 7094 library clock subroutine which has a minimum measurable time increment of 0.6 second.

Accuracy and integration time for transearth case.- The integration times for the transearth trajectory are plotted on figure 4 as a function of A_R for both single and double precision solutions. The position deviations of these solutions from the Cowell results are also presented to give an indication of the trade-off between accuracy and integration time.[4] The velocity deviations in m/sec are nearly the same as the position deviations in km and have been omitted. Note that very good results are obtained for $A_R \leq 10^{-4}$ and the integration time is about 5.5 sec for single precision and 6.5 sec for double precision. Some time can be saved at the expense of errors in the order of 1 or 2 km (and m/sec) by raising A_R as high as 5×10^{-4}. It is doubtful whether one would wish to use higher values of A_R because of the rapid rise in error for a very small decrease in integration time.

Accuracy and integration time for translunar case.- Figure 5 presents the integration time and position and velocity deviations for the translunar trajectories using the same values of A_U, A_L, and A_R as for the data of figure 4. The error in the single precision solution due to round-off is greater than 5 km for all values of A_R. The results for the double precision case are essentially the same as for the transearth trajectory, but the deviation from the Cowell solution for small values of A_R is slightly larger for the translunar case. This result probably arises from the initial velocity being used differently in the two programs. Since the initial velocity is much larger for translunar injection, the last binary digit represents a velocity sufficient to produce position deviations of the order of 0.5 km at the moon. The corresponding deviation for the transearth case is about 1/4 this amount.

[4] The interaction of various parameters and initial conditions sometimes causes the cancellation of errors. Such cancellation produced unrealistically small errors for the two largest values of A_R in the single precision case and for $A_R = 5 \times 10^{-4}$ in the double precision case. For this reason the position deviations given in figure 4 for the single precision cases are the maximum encountered in the data for figures 2 and 3 with A_U set to 0.1, while the one point for double precision is omitted.

It was found that the larger errors in the single precision case arise because near the earth the upper bound on A_Q produces very small step sizes, hence, very small perturbations which are mostly lost during rectification. This difficulty could be remedied by taking several integration steps between rectifications, but it was pointed out earlier that this procedure if done over the entire trajectory requires excessive integration times. Single precision could be used efficiently if the program were made to take several integration steps between rectifications near the centers of attraction and to rectify every integration step elsewhere. The logic and computations for this procedure would require more additional storage than the use of double precision, but there might be a small saving in integration time.

Time requirements for iterative solution of Kepler's equation.- It was pointed out earlier that one of the big advantages of Danby's method over others for space navigation is the ease of restarting the integration. If the integration is to be stopped precisely at a given time in order to process observational data, an additional computation, namely, the inversion of Kepler's equation by iteration, is required. The computation time required for this iteration will be different for each individual solution, but some representative values were obtained as follows: The initial conditions for the translunar trajectory were used as a starting point, and the initial increment of eccentric anomaly was set at 0.1 radian. The program was required to find the change in eccentric anomaly corresponding to a time increment of one hour to an accuracy of one part in 10^7 (i.e., to 0.00036 sec in this case). The time for 100 solutions was measured, and the process was repeated for transearth injection. The same data were also obtained to an accuracy of one part in 10^8 and all the results were presented in table I. The data for the higher accuracy were computed using double precision because when $dt/d\theta$ is small the accuracy of one part in 10^8 cannot be achieved with single precision. The accuracy actually needed remains to be established, but one part in 10^7 should be ample for most applications. Even an additional order of magnitude in accuracy does not require an excessive amount of computation time.

Comparison With Cowell's Method

Two Cowell programs were used for comparison with Danby's method. Cowell program A computes only a single trajectory (no transition matrices) while Cowell program B computes the identical reference trajectory plus the transition matrices. All programming for Danby's method, except for subroutine for multiplying matrices, was done in the Fortran IV computer language but could probably be done more efficiently in assembly language (MAP), commonly called machine language. On the other hand, the Cowell integration subroutine, which comprises nearly half of Cowell program A and over 1/4 of Cowell program B, is written in machine language.

It should be pointed out that the navigation and guidance task requires the integration to be stopped frequently for the processing of observed data and for the computation of several velocity corrections. These operations require the use of the transition matrices so that it is logical to compare Cowell program B, including an appropriate number of restarts, with Danby's method. However, it is of interest to know the requirements of various operations so that data for comparison of the performance of the following tasks have been obtained.

(1) The integration of a single trajectory without intermediate restarts.

(2) The computation of a trajectory and its associated transition matrices without restarts.

(3) The computation of a trajectory and its associated transition matrices with restarts at appropriate times to simulate a navigation problem.

The problem of computing velocity corrections is quite complicated and has been left for a future study except for the following cursory examination: Danby's method inherently includes computing of the two-body transition matrices. These matrices are known to be accurate enough for use in trajectory determination, but introduce large errors in velocity corrections computed at great distances from the terminal point. There are several possible ways of correcting for this inaccuracy, including the use of iteration. Iteration would sacrifice some of the speed advantage of Danby's method over Cowell's, but would eliminate the need for a guidance reference trajectory. (Such a reference trajectory, discussed in reference 7, is correct from injection to the terminal point and should not be confused with the reference conics of Danby's or Encke's method.) Thus, some of the speed of the Danby integration would be traded for a reduction in storage requirements. Other possibilities include the computation of the transition matrices by the method of reference 5, compensation for the effects of perturbations on the transition matrices (ref. 8), or the use of precomputed two-body aim points.

Time requirements.- Cowell program A requires about 18 sec, exclusive of computer output time, to integrate the transearth trajectory and 17 sec in the translunar case or about 2.5 times the requirement for equivalent accuracy using Danby's method. Cowell program B requires about 43 sec for the same integration, that is, about 6.5 times more than is required by Danby's method. The time requirements resulting from a large number of restarts will be discussed later.

Storage requirements.- The program using Danby's method and single precision throughout requires a total of 2775 words of computer storage of which 2398 words are program and 377 are data. These figures do not include the Fortran monitor system or the subroutines for obtaining the ephemerides of the sun and moon. The ephemerides computation, which is the same for both methods of integration, requires about 1700 words of storage. Since it is known that this requirement can be reduced by about an order of magnitude (by use of a more specialized method), it has been eliminated from comparison. When the osculating conic is computed in double precision, the storage requirement increases by 72 data words for a total of 2849.

For comparison, Cowell program A requires 2568 words of storage of which 237 are for data, while Cowell program B requires 4172 words, of which 814 are for data. Thus, when used as part of a navigation system, Danby's method requires about 1/3 less storage than the Cowell program.

Effects of restarting.- Cowell program B and the single precision version of Danby's method were modified slightly to allow the integration to be stopped and restarted. The transearth trajectory was then integrated with each program, the integration being stopped and restarted at 45 times which was considered sufficient for making observations for trajectory estimation. The resulting change in terminal position was 0.02 km for Cowell's method and 0.34 km for Danby's method (using $A_R = 10^{-4}$, $A_U = 0.1$, and $A_L = 0.01$). The integration time for the Cowell program increased to about 2.4 min,[5] while that for Danby's method increased to about 8 sec.

The data in table I indicate that 45 additional solutions of Kepler's equation contribute no more than about 1 sec of the additional integration time for Danby's method. This figure is confirmed when it is noted that the total number of integration steps, including repeats for reducing step size or stopping at the desired time, is increased from 83 to 126. The total number of integration steps could probably be reduced by the use of additional program logic. However, the integration time is still small, and it is doubtful whether this complication is worthwhile unless a considerably greater number of observations is to be made.

Further improvements for use in space navigation systems.- For the sample navigation problem just considered, it has been found that Cowell's method requires about 17.5 times as much computing time as Danby's method. The Cowell program can be modified to reduce this factor to between 8 and 10 at the sacrifice of accuracy. Likewise, the integration time for Danby's method could probably be reduced further by programming improvements including the use of machine language. However, in the particular problem considered, it is known that the integration time required by Danby's method is no greater than the time required to process the data from 45 observations. Further investigation is needed to establish whether the complete navigation system can best be improved by a more refined integration method or by more efficient data processing.

CONCLUDING REMARKS

The data presented here show that for integrating a single reference trajectory and computing the associated transition matrices Danby's method requires only about two-thirds the storage required by Cowell's (program B) method and about one-sixth the computing time. If the transition matrices are omitted, the storage requirement for Cowell's method is slightly less than for

[5] Other Ames programs using the Cowell integration for simulation of guidance and navigation systems (e.g., see ref. 7) use fixed step mode between those observations which are scheduled at reasonably short intervals. These programs require less integration time than the above figure, but the stops for observations result in terminal errors of about 6 km for this number of observations.

Danby's method, but the integration time is still about three times that required for Danby's method. The accuracies of the two methods are (as far as could be ascertained) comparable and are satisfactory for space navigation.

The introduction of a number of stops and restarts to allow for estimation of the trajectory from observed data increases the time required by both methods. However, the increase for Cowell's method is so great that in the sample problem considered, the Cowell integration required 17.5 times as much computing time as Danby's method. The restarts have little effect on the accuracy of either solution, but it is possible to reduce the integration time for Cowell's method by a factor of about 2 at the expense of fairly large terminal errors.

A suitable Encke program was not available for comparison. However, Encke's method requires the solution of the same conic equations, while the computation of the perturbations is more complex than for Danby's method. Also, since Encke's method requires the solution of a set of differential equations for the perturbations, a complex procedure is required to restart the the integration. On this basis, Danby's method can be assumed to be at least competitive with Encke's, particularly if the transition matrices are needed.

Finally, while the equations for the conic solutions and the transition matrices are rather complex, they require only a knowledge of the calculus and elementary matrix theory for their formulation. Furthermore, no detailed knowledge of numerical methods is required to set up the quadrature of perturbing accelerations. For these reasons, Danby's method offers a simplicity and versatility which seems particularly well suited to engineering applications.

Ames Research Center
National Aeronautics and Space Administration
Moffett Field, Calif., June 21, 1966
125-17-05-01-21

APPENDIX A

EQUATIONS FOR THE CONIC TRAJECTORY AND TRANSITION MATRICES

Laplace's f and g formulation, which is summarized below, was used to calculate the reference conic. (See ref. 9 for a detailed explanation.) Following the discussion of the reference conic, the derivation of the equations for the transition matrix is presented. This derivation is intended to correspond as closely as possible with the program listing (SUBROUTINE PARTLS) given in appendix D and for this reason may seem rather awkward from a mathematical point of view. Reference 10 presents different formulations of the equations for the conic trajectory and the transition matrix which are valid for all conics, while the ones given here break down in the parabolic case. The formulas in reference 10 may also require less computer time and storage and their use should be considered in future applications.

The equations for the reference conic are based on the assumption that the position and velocity vectors, $\bar{R}_o$ and $\bar{V}_o$, respectively, at time t_o are given. Then at time t

$$\left.\begin{aligned} \bar{R} &= f\bar{R}_o + g\bar{V}_o \\ \bar{V} &= \dot{f}\bar{R}_o + \dot{g}\bar{V}_o \end{aligned}\right\} \tag{A1}$$

The following set of formulas (eqs. (A2) through (A9)) for the scalars f, g, $\dot{f}$, and $\dot{g}$ and appropriate version of Kepler's equation are given by Pines[1] in an unpublished work.

$$f = 1 - \frac{|a|}{R_o} f_2 \tag{A2}$$

$$g = \Delta t - \sqrt{\frac{|a|^3}{\mu}}\, f_1 \tag{A3}$$

$$\dot{f} = - \frac{\sqrt{\mu|a|}}{R_o R} f_3 \tag{A4}$$

$$\dot{g} = 1 - \frac{|a|}{R} f_2 = 1 - \frac{R_o}{R}(1 - f) \tag{A5}$$

$$\frac{R}{|a|} = f_2 + \frac{R_o}{|a|} f_4 + \frac{\bar{R}_o \cdot \bar{V}_o}{\sqrt{\mu|a|}} f_3 \tag{A6}$$

$$\Delta t = \sqrt{\frac{|a|^3}{\mu}}\left(f_1 + \frac{R_o}{|a|} f_3 + \frac{\bar{R}_o \cdot \bar{V}_o}{\sqrt{\mu|a|}} f_2\right) \tag{A7}$$

[1] Pines, Samuel: Analytic Mechanics Associates, Uniondale, New York.

where μ is the product of the universal gravitation constant and mass of the central body. The semimajor axis, a, is given by

$$\frac{1}{a} = \frac{2}{R_o} - \frac{V_o^2}{\mu}$$

This equation is valid for both elliptic and hyperbolic orbits and gives a negative value for a in the hyperbolic case.

The definitions above are valid for both ellipses and hyperbolas, provided f_i are defined as follows:

(1) For elliptic orbits

$$\left.\begin{aligned} f_1 &= \theta - \sin\theta \\ f_2 &= 1 - \cos\theta \\ f_3 &= \sin\theta \\ f_4 &= \cos\theta \end{aligned}\right\} \qquad \text{(A8)}$$

(2) For hyperbolic orbits

$$\left.\begin{aligned} f_1 &= \sinh\theta - \theta \\ f_2 &= \cosh\theta - 1 \\ f_3 &= \sinh\theta \\ f_4 &= \cosh\theta \end{aligned}\right\} \qquad \text{(A9)}$$

Here θ is the change in eccentric anomaly $(E - E_o)$ or in hyperbolic anomaly $(F - F_o)$, whichever is appropriate.

A series is used for evaluating f_1 and f_2; then f_3 and f_4 can be computed by use of the defining equations. The series for f_1 and f_2 are given by

$$\left.\begin{aligned} f_1 &= \sum_{n=1}^{\infty} \frac{\left(\frac{-a}{|a|}\right)^{n+1} \theta^{2n+1}}{(2n + 1)!} \\ f_2 &= \sum_{m=1}^{\infty} \frac{\left(\frac{-a}{|a|}\right)^{n+1} \theta^{2n}}{2n!} \end{aligned}\right\} \qquad \text{(A10)}$$

The factor $(-a/|a|)$ automatically changes the series from circular to hyperbolic functions. This formulation is similar to the one used in reference 10 (Herrick's variable), but is not valid for the parabolic case.

If equation (A7) is to be solved for the value of θ corresponding to a given Δt, it is necessary to use an iteration. The iteration method used in this study can be understood best by examination of the listing of SUBROUTINE LAPLCE in appendix D and will not be discussed here.

The transition matrices were obtained in closed form by differentiating equations (A1) with respect to the initial components of position and velocity. For convenience, the following notation is defined:

$\left.\begin{matrix} \dfrac{\partial\bar{R}}{\partial\bar{R}_o}, & \dfrac{\partial\bar{R}}{\partial\bar{V}_o}, \\ \dfrac{\partial\bar{V}}{\partial\bar{R}_o}, & \dfrac{\partial\bar{V}}{\partial\bar{V}_o} \end{matrix}\right\}$ 3×3 matrices of first partial derivatives of components of $\bar{R}$ or $\bar{V}$ with respect to the components of $\bar{R}_o$ and $\bar{V}_o$

∇_R gradient operator with the components of position as the independent variable

∇_V gradient operator with the components of velocity as the independent variable

Using these definitions, we can write the transition matrix in partitioned form as

$$\Phi = \begin{vmatrix} \varphi_1 & \varphi_2 \\ \varphi_3 & \varphi_4 \end{vmatrix} = \begin{vmatrix} \dfrac{\partial\bar{R}}{\partial\bar{R}_o} & \dfrac{\partial\bar{R}}{\partial\bar{V}_o} \\ \dfrac{\partial\bar{V}}{\partial\bar{R}_o} & \dfrac{\partial\bar{V}}{\partial\bar{V}_o} \end{vmatrix} \tag{A11}$$

From equation (A1)

$$\left.\begin{aligned} \varphi_1 &= f \quad I + \bar{R}_o(\nabla_R f)^T + \bar{V}_o(\nabla_R g)^T \\ \varphi_2 &= g \quad I + \bar{R}_o(\nabla_V f)^T + \bar{V}_o(\nabla_V g)^T \\ \varphi_3 &= \dot{f} \quad I + \bar{R}_o(\nabla_R \dot{f})^T + \bar{V}_o(\nabla_R \dot{g})^T \\ \varphi_4 &= \dot{g} \quad I + \bar{R}_o(\nabla_V \dot{f})^T + \bar{V}_o(\nabla_V \dot{g})^T \end{aligned}\right\} \tag{A12}$$

where I is a 3×3 unit matrix.

From the definitions of the f_i it can be seen that

$$\left.\begin{aligned} df_1 &= f_2\ d\theta \\ df_2 &= f_3\ d\theta \\ df_3 &= f_4\ d\theta \\ df_4 &= \frac{-a}{|a|}\ f_3\ d\theta \end{aligned}\right\} \tag{A13}$$

Using equations (A13), one can write

$$\nabla f = \frac{-|a|}{R_o}\ f_3 \nabla\theta + \frac{a^2}{R_o{}^2}\ \nabla\ \frac{R_o}{|a|}\ f_2 \tag{A14}$$

$$\nabla g = -\sqrt{\frac{|a|^3}{\mu}}\ f_2 \nabla\theta - f_1 \nabla\left(\sqrt{\frac{|a|^3}{\mu}}\right) \tag{A15}$$

$$\nabla\dot{f} = -\frac{\sqrt{\mu|a|}}{R_o R}\ f_4 \nabla\theta + \frac{\sqrt{\mu|a|}}{R_o{}^2 R}\ f_3 \nabla R_o + \frac{\dot{f}}{\sqrt{\mu|a|}}\ \nabla\left(\sqrt{\mu|a|}\right) - \frac{\dot{f}}{R^2}\ (\nabla\overline{R})^T\overline{R} \tag{A16}$$

$$\nabla\dot{g} = \frac{R_o}{R}\ \nabla f - \frac{1 - f}{RR_o}\ \nabla\overline{R}_o + \frac{R_o}{R^3}\ (1 - f)(\nabla\overline{R}^T)\overline{R} \tag{A17}$$

The notation $\nabla\overline{R}$ is used here to mean $\partial\overline{R}/\partial\overline{R}_o$ or $\partial\overline{R}/\partial\overline{V}_o$ whichever is appropriate. Finally, if Δt is held constant, then

$$\nabla(\Delta t) = 0 = \sqrt{\frac{\mu}{|a|^3}}\ \Delta t \nabla\left(\sqrt{\frac{|a|^3}{\mu}}\right) + \sqrt{\frac{|a|^3}{\mu}}\left(f_2 + \frac{R_o}{|a|}\ f_4 + \frac{\bar{R}_o \cdot \bar{V}_o}{\sqrt{\mu|a|}}\ f_3\right) \nabla\theta$$

$$+ \sqrt{\frac{|a|^3}{\mu}}\ f_3 \nabla\left(\frac{R_o}{|a|}\right) + f_2\sqrt{\frac{|a|^3}{\mu}}\ \nabla\left(\frac{\bar{R}_o \cdot \bar{V}_o}{\sqrt{\mu|a|}}\right)$$

This expression can be solved for $\nabla\theta$ to give

$$\nabla\theta = - \frac{\left\{\sqrt{\frac{\mu}{|a|^3}}\,\Delta t\nabla\left(\sqrt{\frac{|a|^3}{\mu}}\right) + \sqrt{\frac{|a|^3}{\mu}}\left[f_3\nabla\left(\frac{R_o}{|a|}\right) + f_2\nabla\left(\frac{R_o\cdot V_o}{\sqrt{\mu|a|}}\right)\right]\right\}}{\sqrt{\frac{|a|^3}{\mu}}\,\frac{R}{|a|}} \qquad \text{(A18)}$$

For convenience in programming, two one-dimensional arrays labeled Q and C are defined. The part of the Q array which depends only on the values of $\bar{R}_o$ and $\bar{V}_o$ (i.e., not on Δt or θ) is computed in SUBROUTINE CNSTNT while the remainder is computed in SUBROUTINE LAPLCE. These arrays are defined below in tabular form. Note that the elements are given in numerical order and not in the order in which they are computed in program.

Index	Q array	C array
1	f_1	$C_{15}(C_3 - Q_{22}C_{17}Q_2) + Q_{17}Q_3{}^2/Q_9Q_{16}$
2	f_2	$Q_{22}Q_7C_{17}Q_2$
3	f_3	$Q_{17}(Q_3C_{18} - 2C_{17}Q_2)$
4	f_4	$C_{15}C_6 - Q_4Q_7/Q_9Q_{16} - Q_7/Q_{21}$
5	f	$Q_2Q_4Q_{11}/Q_9Q_{15}$
6	g	$Q_4C_{18}/Q_9Q_{13}Q_{16} + Q_7C_{17}$
7	$\dot{f}$	$-Q_7/Q_{15}$
8	$\dot{g}$	$C_{15}C_{10} + C_2$
9	R_o	$Q_{14}Q_{11}Q_{13}Q_2{}^2/Q_{16}$
10	a	$Q_{14}(Q_2C_{18} - 3Q_1C_{17})$
11	$\lvert a\rvert$	$C_1C_{16} - (1 - Q_8)/Q_9Q_{16}$
12	μ	C_2C_{16}
13	$1/\sqrt{\mu\lvert a\rvert}$	C_3C_{16}
14	$\sqrt{\lvert a\rvert^3/\mu}$	$R_o(1 - f)/R^3 = (1 - Q_8)/Q_{15}$
15	R^2	$\mu/R_o{}^3 = Q_{12}/Q_9Q_{21}$
16	R	$R_o/R = Q_9/Q_{16}$
17	$\lvert a\rvert/R_o$	$a/\mu = Q_{10}/Q_{12}$
18	$R_o/\lvert a\rvert$	$Q_{17}C_{16}C_{17}(3Q_1 + Q_{18}Q_3 + 2Q_{19}Q_2)$
19	$(\bar{R}_o\cdot\bar{V}_o)/\sqrt{\mu\lvert a\rvert}$	
20	$V_o{}^2$	
21	$R_o{}^2$	
22	$-a/\lvert a\rvert$	
23	Δt	

It can be shown that formulas for the gradients in equations (A14) through (A18) which apply for both elliptic and hyperbolic orbits are as follows:

$$\nabla_R\left(\frac{R_o}{|a|}\right) = \left(\frac{-2a}{|a|R_o^2} + \frac{1}{|a|R_o}\right)\overline{R}_o = \left(\frac{2Q_{22}}{Q_{21}} + \frac{1}{Q_9Q_{11}}\right)\overline{R}_o$$

$$\nabla_V\left(\frac{R_o}{|a|}\right) = \frac{-2aR_o}{|a|\mu}\overline{V}_o = -2Q_{18}C_{17}\overline{V}_o$$

$$\nabla_R\left(\sqrt{\mu|a|}\right) = \frac{a\sqrt{\mu|a|}}{R_o^3}\overline{R}_o = \frac{Q_{10}}{Q_{13}Q_9Q_{21}}\overline{R}_o$$

$$\nabla_V\left(\sqrt{\mu|a|}\right) = a\sqrt{\frac{|a|}{\mu}}\,\overline{V}_o = Q_{10}Q_{11}Q_{13}\overline{V}_o$$

$$\nabla_R\left(\sqrt{\frac{|a|^3}{\mu}}\right) = \frac{3a}{R_o^3}\sqrt{\frac{|a|^3}{\mu}}\,\overline{R}_o = 3Q_{14}C_{15}C_{17}\overline{R}_o$$

$$\nabla_V\left(\sqrt{\frac{|a|^3}{\mu}}\right) = \frac{3a}{\mu}\sqrt{\frac{|a|^3}{\mu}}\,\overline{V}_o = 3Q_{14}C_{17}\overline{V}_o$$

$$\nabla_R\left(\frac{\overline{R}_o\cdot\overline{V}_o}{\sqrt{\mu|a|}}\right) = \frac{1}{\sqrt{\mu|a|}}\overline{V}_o - \frac{a(\overline{R}_o\cdot\overline{V}_o)}{R_o^3\sqrt{\mu|a|}}\overline{R}_o = Q_{13}\overline{V}_o - Q_{19}C_{15}C_{17}\overline{R}_o$$

$$\nabla_V\left(\frac{\overline{R}_o\cdot\overline{V}_o}{\sqrt{\mu|a|}}\right) = \frac{1}{\sqrt{\mu|a|}}\overline{R}_o - \frac{a(\overline{R}_o\cdot\overline{V}_o)}{\mu\sqrt{\mu|a|}}\overline{V}_o = Q_{13}\overline{R}_o - Q_{19}C_{17}\overline{V}_o$$

$$\nabla_R\theta = -\frac{a}{R_o^3}\frac{|a|}{R}\left(3\sqrt{\frac{\mu}{|a|^3}}\,\Delta t - \frac{2R_o}{|a|}f_3 - \frac{\bar{R}_o \cdot \bar{V}_o}{\sqrt{\mu|a|}}f_2 + \frac{R_o^2}{a|a|}f_3\right)\bar{R}_o$$

$$-\frac{1}{\mu|a|}\frac{|a|}{R}f_2\bar{V}_o$$

$$= -\left(C_{15}C_{18} + \frac{Q_3}{Q_9Q_{16}}\right)\bar{R}_o - \frac{Q_{11}Q_{13}Q_2}{Q_{16}}\bar{V}_o$$

$$\nabla R^{\theta} = -\frac{a}{\mu}\frac{|a|}{R}\left(3\sqrt{\frac{\mu}{|a|^3}}\,\Delta t - \frac{2R_o}{|a|}f_3 - \frac{\bar{R}_o \cdot \bar{V}_o}{\sqrt{\mu|a|}}f_2V_o - \frac{1}{\sqrt{\mu|a|}}\frac{|a|}{R}f_2\right)\bar{V}_o$$

$$= -C_{18}\bar{V}_C - \frac{Q_{11}Q_3Q_2}{Q_{16}}\bar{R}_o$$

Substitution of the gradients into equations (A14) through (A17) gives

$$\nabla_R f = -Q_{17}Q_3\nabla_R\theta - 2Q_{17}C_{15}C_{17}Q_2\bar{R}_o + \frac{Q_{17}Q_2}{Q_{21}}\bar{R}_o$$

$$= C_1\bar{R}_o + C_2\bar{V}_o$$

$$\nabla_V f = -Q_{17}Q_3\nabla_V\theta + 2Q_{17}{}^2Q_2Q_{18}C_{17}\bar{V}_o = C_2\bar{R}_o + C_3\bar{V}_o$$

$$\nabla_R g = -Q_{14}Q_2\nabla_R\theta - 3Q_1Q_{14}C_{15}C_{17}\bar{R}_o = C_8\bar{R}_o + C_9\bar{V}_o$$

$$\nabla_V g = -Q_{14}Q_2\nabla_V\theta - 3Q_{14}C_{17}Q_1\bar{V}_o = C_9\bar{R}_o + C_{10}\bar{V}_o$$

$$\nabla_R \dot{f} = \frac{-Q_4}{Q_{13}Q_9Q_{16}} \nabla_R \theta - \frac{Q_7}{Q_9} \bar{R}_o + Q_7C_{15}C_{17}\bar{R}_o + C_7\left(\frac{\partial \bar{R}}{\partial \bar{R}_o}\right)^T \bar{R}$$

$$= C_4\bar{R}_o + C_5\bar{V}_o + C_7\varphi_1^T\bar{R}$$

$$\nabla_V \dot{f} = \frac{-Q_4}{Q_{13}Q_9Q_{16}} \nabla_V \theta + Q_7C_{17}\bar{V}_o + C_7\varphi_2^T\bar{R} = C_5\bar{R}_o + C_6\bar{V}_o + C_7\varphi_2^T\bar{R}$$

$$\nabla_R \dot{g} = C_{16}\nabla_R f - \frac{(1 - Q_8)}{Q_9Q_{16}} \bar{R}_o + C_{14}\varphi_1^T\bar{R} = C_{11}\bar{R}_o + C_{12}\bar{V}_o + C_{14}\varphi_1^T\bar{R}$$

$$\nabla_V \dot{g} = C_{16}\nabla_V f + C_{14}\varphi_2^T\bar{R} = C_{12}\bar{R}_o + C_{13}\bar{V}_o + C_{14}\varphi_2^T\bar{R}$$

Finally, substitution in equations (A2) gives

$$\varphi_1 = fI + \bar{R}_o(\nabla_R f)^T + \bar{V}_o(\nabla_R g)^T$$

$$= fI + \bar{R}_o(C_1\bar{R}_o^T + C_2\bar{V}_o^T) + \bar{V}_o(C_8\bar{R}_o^T + C_9\bar{V}_o^T)$$

$$= fI + (C_1\bar{R}_o + C_8\bar{V}_o)\bar{R}_o^T + (C_2\bar{R}_o + C_9\bar{V}_o)\bar{V}_o^T$$

$$\varphi_2 = gI + \bar{R}_o(\bar{\nabla}_V f)^T + \bar{V}_o(\bar{\nabla}_V g)^T$$

$$= gI + (C_2\bar{R}_o + C_9\bar{V}_o)\bar{R}_o^T + (C_3\bar{R}_o + C_{10}\bar{V}_o)\bar{V}_o^T$$

$$\varphi_3 = \dot{f}I + \bar{R}_o(\nabla_R \dot{f})^T + \bar{V}_o(\nabla_R \dot{g})^T$$

$$= \dot{f}I + C_4\bar{R}_o\bar{R}_o^T + C_5\bar{R}_o\bar{V}_o^T + C_{11}\bar{V}_o\bar{R}_o^T + C_{12}\bar{V}_o\bar{V}_o^T$$

$$+ C_7\bar{R}_o\bar{R}^T\varphi_1 + C_{14}\bar{V}_o\bar{R}^T\varphi_1$$

$$= fI + (C_4\bar{R}_o + C_{11}\bar{V}_o)\bar{R}_o^T + (C_5\bar{R}_o + C_{12}\bar{V}_o)\bar{V}_o^T$$

$$+(C_7\bar{R}_o + C_{14}\bar{V}_o)\bar{R}^T\varphi_1$$

$$
\begin{aligned}
\varphi_4 &= \dot{g}I + \overline{R}_o(\nabla_V \dot{f})^T + \overline{V}_o(\nabla_V \dot{g})^T \\
&= \dot{g}I + C_5\overline{R}_o\overline{R}_o^T + C_6\overline{R}_o\overline{V}_o^T + C_{12}\overline{V}_o\overline{R}_o^T + C_{13}\overline{V}_o\overline{V}_o^T \\
&\quad + C_7\overline{R}_o\overline{R}^T\varphi_2 + C_{14}\overline{V}_o\overline{R}^T\varphi_2 \\
&= \dot{g}I + (C_5\overline{R}_o + C_{12}\overline{V}_o)\overline{R}_o^T + (C_6\overline{R}_o + C_{13}\overline{V}_o)\overline{V}_o^T + (C_7\overline{R}_o + C_{14}\overline{V}_o)\overline{R}^T\varphi_2
\end{aligned}
$$

APPENDIX B

EQUATIONS FOR PERTURBING ACCELERATIONS

The perturbing accelerations are computed on the basis of the four-body equations of motion given in reference 7. The coordinate system is Cartesian with the X axis positive toward the vernal equinox, the Z axis parallel to the earth's polar axis, and the Y axis completing a right handed orthogonal set. When the vehicle is within 66,000 km of the moon a selenocentric system is used while otherwise it is geocentric. The vehicle's equations of motion in the geocentric system are as follows:

$$\ddot{X} = -\frac{\mu_e X}{R_e{}^3}\left[1 + J\left(\frac{R_Q}{R_e}\right)^2\left(1 - \frac{5Z^2}{R_e{}^2}\right)\right] - \frac{\mu_m}{\nabla_m{}^3}(X - X_M) - \frac{\mu_m X_m}{R_m{}^3} - \frac{\mu_s(X - X_s)}{s^3} - \frac{\mu_s X_s}{R_e{}^2}$$

$$\ddot{Y} = -\frac{\mu_e Y}{R_e{}^3}\left[1 + J\left(\frac{R_Q}{R_e}\right)^2\left(1 - \frac{5Z^2}{R_e{}^2}\right)\right] - \frac{\mu_m}{\Delta_m{}^3}(Y - Y_m) - \frac{\mu_m Y_m}{R_m{}^3} - \frac{\mu_s(Y - Y_s)}{\Delta_s{}^3} - \frac{\mu_s Y_s}{R_s{}^3}$$

$$\ddot{Z} = -\frac{\mu_e Z}{R_e{}^3}\left[1 + J\left(\frac{R_Q}{R_e}\right)^2\left(3 - \frac{5Z^2}{R_e{}^2}\right)\right] - \frac{\mu_m(Z - Z_m)}{\Delta_m{}^3} - \frac{\mu_m Z_m}{R_m{}^3} - \frac{\mu_s(Z - Z_s)}{\Delta_s{}^3} - \frac{\mu_s Z_s}{R_s{}^3}$$

where (X_m, Y_m, Z_m) and (X_s, Y_s, Z_s) are the positions of the moon and sun, respectively. Other quantities are defined as follows:

$$R_e = \sqrt{X^2 + Y^2 + Z^2}$$

$$R_m = \sqrt{X_m{}^2 + Y_m{}^2 + Z_m{}^2}$$

$$R_s = \sqrt{X_s{}^2 + Y_s{}^2 + Z_s{}^2}$$

$$\Delta_m = \sqrt{(X - X_m)^2 + (Y - Y_m)^2 + (Z - Z_m)^2}$$

$$\Delta_s = \sqrt{(X - X_s)^2 + (Y - Y_s)^2 + (Z - Z_s)^2}$$

$$\mu_e = 3.986031\text{x}10^5 \text{ km}^3/\text{sec}^2$$

$$\mu_m = 4.8938269\text{x}10^3 \text{ km}^3/\text{sec}^2$$

$$\mu_s = 1.3255\times10^{11}\ \text{km}^3/\text{sec}^2$$

$$JR_Q^2 = 6.6043958\times10^4\ \text{km}^2$$

where R_Q is the mean equatorial radius of the earth.

The three components of the vector $\bar{u}$ of perturbing accelerations (eq. (2)) are obtained by subtracting the first harmonic term (the two-body portion) of the acceleration from the total. Thus, in geocentric coordinates:

$$u_4 = \ddot{X} + \frac{\mu_e X}{R_e^3}$$

$$u_5 = \ddot{Y} + \frac{\mu_e Y}{R_e^3}$$

$$u_6 = \ddot{Z} + \frac{\mu_e Z}{R_e^3}$$

In selenocentric coordinates, the terms involving J are omitted and X, Y, and Z now denote the vehicle's position with respect to the center of the moon. In this case

$$u_4 = \ddot{X} + \frac{\mu_m X}{\Delta_m^3}$$

$$u_5 = \ddot{Y} + \frac{\mu_m Y}{\Delta_m^3}$$

$$u_6 = \ddot{Z} + \frac{\mu_m Z}{\Delta_m^3}$$

(Note that in the program listings the position vectors of the moon $\overline{PM}$ and the sun $\overline{PS}$ and the velocity vector of the moon $\overline{VM}$ always refer to geocentric coordinates.)

APPENDIX C

SIMPSON'S RULE FOR UNEQUAL TIME INCREMENTS

It is desired to integrate the transformed forcing function y over the time interval from t_o to t_2. We assume that each component y_i can be fit over the interval with an equation of the form

$$y = a_2(t - t_o)^2 + a_1(t - t_o) + a_o \tag{C1}$$

The subscript, i, indicating the component of y has been omitted and equation (C1) may be taken as representing any of the six components of y.

Let t_1 be some time between t_o and t_2 and define

$$\left.\begin{aligned} h &= t - t_o \\ h_1 &= t_1 - t_o \\ h_2 &= t_2 - t_o \\ y_o &= y(t_o) \\ y_1 &= y(t_1) \\ y_2 &= y(t_2) \end{aligned}\right\} \tag{C2}$$

then equations (C1) and (C2) can be combined to give

$$\left.\begin{aligned} y_o &= a_o \\ y_1 &= a_2h_1^2 + a_1h_1 + a_o \\ y_2 &= a_2h_2^2 + a_1h_2 + a_o \end{aligned}\right\} \tag{C3}$$

The coefficients of t in equation (C1) can be found by the simultaneous solution of equations (C3), and are

$$\left.\begin{aligned}
a_o &= y_o \\
a_1 &= \frac{(y_1 - y_o)h_2^2 - (y_2 - y_o)h_1^2}{h_1h_2^2 - h_1^2h_2} \\
a_2 &= \frac{(y_2 - y_o)h_1 - (y_1 - y_o)h_2}{h_1h_2^2 - h_1^2h_2}
\end{aligned}\right\} \qquad \text{(C4)}$$

The integral of equation (C1) is

$$\int_{t_o}^{t_2} y\,dt = \int_o^{h_2} (a_2h^2 + a_1h + a_o)dh = \frac{a_2}{3} h_2^3 + \frac{a_1}{2} h_2^2 + a_oh_2 \qquad \text{(C5)}$$

Substitution for the a_i from equations (C4) gives

$$\int_{t_o}^{t_2} y\,dt = \frac{h_2^3}{3}\left[\frac{(y_2 - y_o)h_1 - (y_1 - y_o)h_2}{h_1h_2^2 - h_1^2h_2}\right] + \frac{h_2^2}{2}\left[\frac{(y_1 - y_o)h_2^2 - (y_2 - y_o)h_1^2}{h_1h_2^2 - h_1^2h_2}\right]$$

$$+ y_oh_2 \qquad \text{(C6)}$$

Collecting terms in y_n changes equation (C6) to

$$\int_{t_o}^{t_2} y\,dt = \frac{h_2h_1(2h_2 - 3h_1)y_2 + h_2^3y_1 + h_2(3h_1 - h_2)(h_2 - h_1)y_o}{6h_1(h_2 - h_1)} \qquad \text{(C7)}$$

The coefficients of the y_n are functions only of time and are valid for each component of y. They are stored in the AI array in the program.

APPENDIX D

PROGRAM LISTINGS

The following pages present the Fortran IV listings of the main program and subroutines used for Danby's integration method. The single precision version is presented here, but the program is set up for easy conversion to double precision computation of the osculating conic, to more than one integration step between rectifications, and for inclusion in a complete guidance and navigation program. For this reason all common data are stored in "labeled common," and storage is provided for a number of superfluous variables. Comment cards have been inserted at appropriate points to explain the operation of the program.

In addition to the subroutines for which complete listings are provided, the program uses five general purpose subroutines. Only enough listing is presented to explain the use of these subroutines.

```
$IBFTC TG0503  NOREF
CTG0503 DANBY INTEGRATION PROGRAM              J D MCLEAN
C USES SUBROUTINES TG050B, D, S, T, U, V, W, X, AND Y
      DIMENSION GP(6),ED(6),DELR(3),DELV(3),GC(6,3),DELX(6),XPREV(6),
     1DELYP(6),DD(6),AI(3),DELY(6),H(3),AP(6,6),X(6),     XPRES(6),
     2XZERO(6),KK(6),VARLST(14),TX(3) ,RR(3),VV(3),AE(6,6)
      COMMON /DNBY/ RZERO(3),VZERO(3),PM(3),VM(3),Q(23),PS(3),G(3),K(6),
     1ACCREC,ACCUI,ACCLI,US,UE,UM,CJ2,DELM,R,V,DELS,DM(3),TMSTOP,
     2TIMEA,TIME,CHOVER,HT,HE,RPRES(3),VPRES(3),ESTOP,DELTE,AA(3,3,5),
     3IPRINT,AO(6,6),ICYCLE,IMOON,ISUN,TMOON(32,25),TSUN(4,16),DE(3)
      EQUIVALENCE (XZERO(1),RZERO(1)),(XZERO(4),VZERO(1)),(DELX(1),DELR
     1(1)),(DELX(4),DELV(1)),(XPRES(1),RPRES(1)),(XPRES(4),VPRES(1))

C     ACCUMULATE TOTAL INTEGRATION TIME IN DOUBLE PRECISION
      DOUBLE PRECISION TAPRS,TADP

C     DEFINITION OF VARIABLES
C     TS=STOP TIME
C     TA=TIME FROM INJECTION TO PRESENT
C     UE,UM,US = MU OF EARTH,SUN,MOON
C     HT=+1.0 FOR FORWARD INTEGRATION, -1.0 FOR BACKWARD
C     HE= STARTING INCREMENT IN ECCENTRIC ANOMALY
C     RR= POSITION VECTOR
C     VV= VELOCITY VECTOR
C     K(1)=-1 FOR MOON CENTERED COORDINATES, +1 FOR EARTH CENTERED
C     K(2) NOT USED
C     K(3)= LIMIT ON NUMBER OF PASSES THROUGH INTEGRATION LOOP
C     K(4)= ZERO EXCEPT WHEN COORDINATES ARE TO BE SWITCHED
C     K(5)= ZERO EXCEPT WHEN TRAJECTORY IS COMPLETED
C     K(6) NOT USED
C     DATE= JULIAN DATE OF START OF SUN-MOON TABLES
C     DAYS= NOT USED
C     ACCREC= A SUB R -- SEE TEXT
C     ACCUI= A SUB U
C     ACCLI= A SUB L
C     CJ2= J*(R SUB Q)**2 -- SEE APPENDIX B
C     CHOVER = RANGE FROM MOON OF COORDINATE SWITCH
C     TI= INJECTION TIME
C     PS= POSITION VECTOR OF SUN
C     PM= POSITION VECTOR OF MOON
```

```
C     DELTE IS TOTAL INCREMENT IN ECCENTRIC ANOMALY FROM LAST
C       RECTIFICATION
C     VM=VELOCITY VECTOR OF MOON
C     DELM= VEHICLE-MOON DISTANCE
C     AE IS TRANSITION MATRIX FROM START OF TRAJECTORY TO PRESENT

      CALL CLOCK(TX)

C      FROM HERE TO STATEMENT 100 INITIALIZES PROGRAM FOR ONE TRAJECTORY
C     INTEGRATION

  200 READ(5,111) TS,TA,UEI,UMI,HTI,HEI,(RR(I),I=1,3),(VV(I),I=1,3)
  111 FORMAT(3E16.8)
      READ(5,901) (K(J),J=1,6)
      READ(5,111) DATE,DAYS,TIME,ACCREC,ACCUI,ACCLI,CJ2,US,CHOVER
  901 FORMAT(6I5)
      TMSTOP=TS
      TIMEI=TIME-TA
      TIMEA=TA
      TAPRS=TA
      TADP=TA
      UE=UEI
      UM=UMI
      HT=HTI
      HE=HEI
      DO 810 I=1,3
      RZERO(I)=RR(I)
  810 VZERO(I)=VV(I)
      WRITE(6,902)TMSTOP,TIMEA,UE
  902 FORMAT(1H1,7HTMSTOP=,D25.16,6HTIMEA=,D25.16,3HHE=,D25.16)
      WRITE(6,903) UM,HT,HE
  903 FORMAT(1H ,3HUM=,D25.16,3HHT=,D25.16,3HHE=,D25.16)
      WRITE(6,904) (RZERO(I),I=1,3)
  904 FORMAT(1H ,2HX=,D25.16,2HY=,D25.16,2HZ=,D25.16)
      WRITE(6,905)(VZERO(I),I=1,3)
  905 FORMAT(1H ,3HXD=,D25.16,3HYD=,D25.16,3HZD=,D25.16)
      WRITE(6,906)(K(I),I=1,6)
  906 FORMAT(1H0,2HK=,6I5)
      WRITE(6,118)
  118 FORMAT(1H-,5X,5HTIMEA,12X,5HX(KM),12X,5HY(KM),12X,5HZ(KM),10X,10HX
     1D(KM/SEC),7X,10HYD(KM/SEC),7X,10HZD(KM/SEC),5X,9HSTEP SIZE)
      WRITE(6,119)
  119 FORMAT(1H ,6X,4HDELX,13X,4HDELY,13X,4HDELZ,13X,4HDELR,14X,1HR,16X,
     11HV,15X,4HDELM)
      CALL MSLOAD(TMOON(1,1),32,TSUN(1,1),4,DATE,NOMPTS,NOSPTS)
      CALL CLOCK(TX(2))
      IMOON=0
      ISUN=0
      KERR=0
      HEP=HE
      ICYCLE=0
      DO 10 I=1,36
   10 AE(I,1)=0.0
      DO 11 I=1,6
      DELX(I)=0.0
   11 AE(I,I)=1.0

C     BEGINNING OF INTEGRATION LOOP
  100 DO 7 I=1,3
    7 RPRES(I)=RZERO(I)
      CALL MOON(TIME,PM(1),TMOON(1,1),32,NOMPTS,KERR,3HPOS,IMOON)
    9 CALL SUNP(TIME,PS(1),TSUN(1,1),4,NOSPTS,KERR,ISUN)

    8 CALL GVEC
C     GVEC COMPUTES FORCING FUNCTION

      DELMP=DELM
      V=0.0
      DO 6 I=1,3
    6 V=V+VZERO(I)**2
      V=SQRT(V)
C     OUTPUTS TIME IN DAYS
      TIMEB=TIMEA/86400.0
  103 FORMAT(1H0,E16.8,6E17.8,E13.4/7E17.8,I5)
      IF(K(5)) 221,221,200
```

```
C      CNSTNT COMPUTES PART OF Q ARRAY AFTER RECTIFICATION
   221 CALL CNSTNT

       DO 21 I=1,3
    21 GC(I,1)=0.0

       DELTE=0.0
       ESTOP=1.0
       TPRS=TIME
       TAPRS=TADP
   212 MA=1
    12 ICYCLE=ICYCLE+1
       IF(ICYCLE-K(3)) 110,200,200
   110 GO TO (24,17,17),MA
    24 K(5)=0
C      EACH TRANSFORMED FORCING FUNCTION, Y IN TEXT, IS ONE COLUMN OF GC
C      STORE INITIAL VALUE OF G IN FIRST COLUMN OF GC

       DO 25 I=1,3
    25 GC(I+3,1)=G(I)
   241 MA=2
       GO TO 12

C      INCREMENT DELTE ONE STEP
    17 DELTE=DELTE+HE

C      COMPUTE NEW 2 BODY POSITION, RPRES, AND VELOCITY, VPRES
       CALL LAPLCE

C      COMPUTE TRANSITION MATRIX, AO, FROM RECTIFICATION TO PRESENT
       CALL PARTLS

       H(MA-1)=Q(23)
       TAPRS=TADP+Q(23)
       TPRS=TAPRS+TIMEI
       CALL SUNP(TPRS,PS(1),TSUN(1,1),4,NOSPTS,KERR,ISUN)
    19 CALL MOON(TPRS,PM(1),TMOON(1,1),32,NOMPTS,KERR,3HPOS,IMOON)

C      COMPUTE NEW FORCING FUNCTION
    20 CALL GVEC
```

```
C     TRANSFORM G TO LAST RECTIFICATION TIME AND STORE IN GC
      CALL XPYXM(G(1),AA(1,1,1),GC(4,MA),1,3,3,3,1)

      DO 22 I=1,3
   22 G(I)=-G(I)
      CALL XPYXM(G(1),AA(1,1,2),GC(1,MA),1,3,3,3,1)

C     TEST WHETHER GC ARRAY HAS BEEN FILLED
      GO TO(200,230,13),MA

  230 MA=3
      GO TO 12
   13 IF(K(5).GT.0) GO TO 30

C     TEST WHETHER STOP TIME HAS BEEN EXCEEDED, IF SO
  213 TST=HT*(TMSTOP-TAPRS)
      IF(TST) 14,18,30

C     COMPUTE EXACT VALUE OF HE TO STOP AT CORRECT TIME
   14 Q(23)=TMSTOP-TIMEA
      ESTOP=0.0
      DELTE=0.0
      CALL LAPLCE
      HE=(ESTOP-DELTE)/2.0
      K(5)=1
      GO TO 241
   18 K(5)=1

C     COMPUTE AI ARRAY -- SEE APPENDIX C
   30 SS=H(2)-H(1)
      AI(1)=   SS*H(2)*(3.0*H(1)-H(2))
      AI(2)=H(2)**3
      AI(3)=H(1)*H(2)*(2.0*H(2)-3.0*H(1))
      SS=SS*H(1)*6.0
      DO 29 I=1,3
   29 AI(I)=AI(I)/SS
      SS=H(2)/2.0
```

```
C     GP= INTEGRAL OF TRANSFORMED FORCING FUNCTION
      CALL XPYXM(GC(1,1),AI(1),GP(1),6,3,3,1,1)

C     COMPUTE DG -- DG= A SUB Q IN TEXT
      DO 229 I=1,6
      ED(I)=GP(I)-(GC(I,1)+GC(I,3))*SS
  229 DD(I)=GP(I)-H(2)*GC(I,1)
      CALL ROOT(DD(1),DR2,DDM,1)
      CALL ROOT(DD(1),DV2,DDM,4)
      DG=0.0
      DO 40 I=1,3
   40 DG=DG+(ED(I)**2)/DR2+(ED(I+3)**2)/DV2
      TST5=DG-ACCUI
      TST6=DG-ACCLI
      IF(TST5.GT.0.0) GO TO 41
   43 DO 243 I=1,6

C      DELX IS USED ONLY WHEN NOT RECTIFYING EVERY STEP
  243 GP(I)=GP(I)+DELX(I)

C     DELY= PERTURBATION STATE VECTOR AT PRESENT
      CALL XPYXM(AO,GP,DELY,6,6,6,1,1)

C     CHECK CONSTRAINTS ON MAGNITUDE OF PERTURBATION
      CALL ROOT(DELY(1),DRSQ,RDEL,1)
  449 IF(R-DELM) 450,450,451
  450 TST7=RDEL/R-ACCREC
      TST8=RDEL/R-0.2*ACCREC
      GO TO 452
  451 TST7=RDEL/DELM-ACCREC
      TST8=RDEL/DELM-0.2*ACCREC
  452 IF(TST7.LE.0.0) GO TO 247

C     THIS SECTION USED WHEN STEP SIZE MUST BE REDUCED
   41 HE=HE/2.0
  251 K(5)=0
      DELTE=0.0
      GO TO 241
```

```
C     UPDATE TIME AND AE AFTER RECTIFICATION
  250 K(6)=1
      DO 51 I=1,6
   51 DELX(I)=0.0
      TIME=TPRS
      TADP=TAPRS
      TIMEA=TAPRS
   60 CALL XPYXM(AO,AE,AA,6,6,6,6,1)
      DO 61 I=1,36
   61 AE(I,1)=AA(I,1,1)

C     RETURN TO BEGINNING OF LOOP UNLESS TRAJECTORY IS TO BE TERMINATED
   62 IF(K(5).LE.0) GO TO 100

      CALL CLOCK(TX(3))
      WRITE(6,2) (TX(I),I=1,3),ICYCLE
    2 FORMAT(1H0,17HEND OF TRAJECTORY,5X,2HTX,3E17.8,5X,6HICYCLE,I5)

      WRITE (6,103)TIMEB,(XZERO(J),J=1,6),HE,(DELY(J),J=1,3),RDEL,R,V,DE
     1LM,ICYCLE
      GO TO 100


C     CHECK FOR COORDINATE SWITCH AND ADJUST HE SO THAT DELM WILL LIE
C      WITHIN 5000 KM OF CHOVER
  247 TST2=K(1)
      TST2=TST2*(DELM-CHOVER)
      K(6)=0
      TST3=ABS(TST2)- 5000.0
      IF(TST2.LT.0.0) GO TO 39
      IF(TST3.GT.0.0) GO TO 37
   33 SHFT=K(1)
      GO TO 38
   37 RATIO=ABS(DELMP-CHOVER)/ABS(DELMP-DELM)
      HE=HE*RATIO
      GO TO 251
   38 K(4)=1
      GO TO 47
```

```
C       CHECK WHETHER STEP SIZE SHOULD INCREASE
   39 IF(TST6.GT.0.0) GO TO 47
      IF(TST8.GT.0.0) GO TO 47
   45 HE=HE*2.0

C       THIS SECTION RECTIFIES CONIC AND SWITCHES COORDINATES IF NEEDED
   47 DO 48 I=1,6
   48 XZERO(I)=XPRES(I)+DELY(I)
      DELMP=DELM
      IF(K(4)) 250,250,52
   52 CALL MOON(TPRS,VM(1),TMOON(1,1),32,NOMPTS,KERR,3HVEL,IMOON)
      IF(KERR.NE.0) GO TO 200
   53 DO 54 I=1,3
      XZERO(I)=XZERO(I)+SHFT*PM(I)
   54 XZERO(I+3)=XZERO(I+3)+SHFT*VM(I)
      K(4)=0
      IF(SHFT.GT.0.0) GO TO 56
   55 K(1)=1
      WRITE (6,102)
  102 FORMAT(1H0,59HTHE FOLLOWING SETS OF DATA ARE IN MOON CENTERED COOR
     1DINATES)
      GO TO 57
   56 K(1)=-1
      WRITE (6,101)
  101 FORMAT(1H0,60HTHE FOLLOWING SETS OF DATA ARE IN EARTH CENTERED COO
     1RDINATES)
   57 HE=HEP
      GO TO 250
      END
```

```
$IBFTC TG050V  NOREF
C      SUBROUTINE LAPLCE  J D MCLEAN  COMPUTES TWO BODY POSITION AND
C     VELOCITY
      SUBROUTINE LAPLCE
      COMMON /DNBY/ RZERO(3),VZERO(3),PM(3),VM(3),Q(23),PS(3),G(3),K(6),
     1ACCREC,ACCUI,ACCLI,US,UE,UM,CJ2,DELM,R,V,DELS,DM(3),TMSTOP,
     2TIMEA,TIME,CHOVER,HT,HE,RPRES(3),VPRES(3),ESTOP,DELTE,AA(3,3,5),
     3IPRINT,AO(6,6),ICYCLE,IMOON,ISUN,TMOON(32,25),TSUN(4,16),DE(3)
C     E IS INCREMENT IN ECCENTRIC ANOMALY
      E=DELTE
      EP=DELTE
      DTP=0.0
      EINC=HE

      DO 15 I=1,20

C     IF ESTOP IS ZERO KEPLERS EQUATION IS SOLVED TO GET DELTE FOR STOP
C     AT CORRECT TIME, OTHERWISE OUTER LOOP IS OMITTED
      IF(ESTOP) 9,8,9

    8 E=EP+EINC

C     KD=0 UNLESS EINC IS TO BE SUBTRACTED FROM E
      KD=0
   10 IF(E-EP) 9,14,9
    9 Q(1)=0.0
      Q(2)=0.0

C     S IS A DUMMY VARIABLE
      S=1.0

C     THIS LOOP SOLVES SERIES FOR Q(1) AND Q(2) -- SEE APPENDIX A
      DO 11 J=1,50
      RN=J
      RN=2.0*RN
      S= S*Q(22)*E**2/(RN*(RN-1.0))
      H=Q(22)*S
      QZ=Q(2)
      Q(2)=Q(2)+H
   41 Q(1)=Q(1)+H*E/(RN+1.0)
```

```
C     SERIES IS CARRIED OUT UNTIL LAST TERM DOES NOT CHANGE Q(2)
      IF(Q(2)-QZ) 11,12,11
   11 CONTINUE
      WRITE (6,101)S
  101 FORMAT(2H0,38HQ1-Q2 SERIES HAS FAILED TO CONVERGE S=,E15.8)


C     COMPUTE TIME INCREMENT
   12 Q(4)=1.0+Q(2)*Q(22)
      Q(3)=E+Q(1)*Q(22)
   42 S=Q(14)*(Q(1)+Q(18)*Q(3)+Q(19)*Q(2))

C     IF NOT ITERATING FOR STOP TIME LEAVE LOOP
      IF(ESTOP) 14,23,14

C     TEST WHETHER TIME INCREMENT HAS BEEN FOUND TO 7 PLACES, IF NOT
   23 TST1= ABS(1.0-S/Q(23))-.1E-6
      IF(TST1) 14,14,13

C     TEST WHETHER TIME INCREMENT IS TOO LARGE, IF NOT
   13 TST2= ABS(S)-ABS(Q(23))
      IF(TST2)30,30,31

C     INCREMENT E AND REPEAT ITERATION
   30 EP=E
      DTP= ABS(S)
      GO TO 15

C     IF E IS TOO LARGE AND KD=0 COMPUTE RATIO OF ACTUAL TIME INCREMENT
C     TO DESIRED ONE
   31 IF(KD) 34,34,38
   34 RATIO=( ABS(Q(23))-DTP)/( ABS(S)-DTP)

C     IF RATIO IS LESS THAN 0.1 MULTIPLY EINC BY 0.1 AND START AT
C     BEGINNING OF LAST INTERVAL
      IF(RATIO-0.1) 35,35,36
   35 EINC=EINC*0.1
      GO TO 8
```

```
C     IF RATIO IS GREATER THAN 0.9 MULTIPLY EINC BY 0.1, SET KD=1 AND
C     BEGIN REDUCING E
   36 IF(RATIO-0.9) 39,37,37
   37 EINC=EINC*0.1
   38 E=E-EINC
      KD=1
      GO TO 10

C     IF RATIO LIES BETWEEN 0.1 AND 0.9 USE LINEAR INTERPOLATION
   39 TINC=TINC*RATIO
      GO TO 8
   15 E=E
      WRITE (6,102)TST1
  102 FORMAT(1H0,43HTIME ITERATION HAS FAILED TO CONVERGE TST1=,E15.8)

C     IF ITERATING FOR STOP TIME RETURN TO MAIN PROGRAM
   14 IF(ESTOP) 25,24,25
   24 ESTOP=E
      GO TO 33

C     COMPUTE NEW POSITION AND VELOCITY
   25 Q(23)=S
      ROA=Q(2)+Q(18)*Q(4)+Q(19)*Q( 3)
      Q(5)=1.0-Q(17)*Q(2)
      Q(6)=Q(23)-Q(14)*Q(1)
      Q(7)=-Q(17)*Q(3)/(ROA*Q(14))
      Q(8)=1.0-Q(2)/ROA
      Q(16)=ROA*Q(11)
   43 Q(15)=Q(16)**2
  100 DO 16 I=1,3
      RPRES(I)=Q(5)*RZERO(I)+Q(6)*VZERO(I)
   16 VPRES(I)=Q(7)*RZERO(I)+Q(8)*VZERO(I)
   33 RETURN
      END
```

```
$IBFTC TG050W  NOREF
C       SUBROUTINE CNSTNT  J D MCLEAN COMPUTES CONSTANTS FOR LAPLCE
C
      SUBROUTINE CNSTNT
      COMMON /DNBY/ RZERO(3),VZERO(3),PM(3),VM(3),Q(23),PS(3),G(3),K(6),
     1ACCREC,ACCUI,ACCLI,US,UE,UM,CJ2,DELM,R,V,DELS,DM(3),TMSTOP,
     2TIMEA,TIME,CHOVER,HT,HE,RPRES(3),VPRES(3),ESTOP,DELTE,AA(3,3,5),
     3IPRINT,AO(6,6),ICYCLE,IMOON,ISUN,TMOON(32,25),TSUN(4,16),DE(3)
      IF(K(1)) 17,17,18
   17 Q(12)=UE
      GO TO 19
   18 Q(12)=UM
   19 Q(19)=0.0
      Q(20)=0.0
      Q(21)=0.0
      DO 10 I=1,3
      Q(19)=Q(19)+RZERO(I)*VZERO(I)
      Q(20)=Q(20)+VZERO(I)**2
   10 Q(21)=Q(21)+RZERO(I)**2
      Q(9)= SQRT(Q(21))
      Q(1)=(2.0/Q(9))-(Q(20)/Q(12))
      Q(10)=1.0/Q(1)
      Q(11)= ABS(Q(10))
      Q(13)=1.0/( SQRT(Q(11)*Q(12)))
      Q(22)=-Q(10)/Q(11)
      Q(17)=Q(11)/Q(9)
      Q(18)=1.0/Q(17)
      Q(14)=Q(13)*Q(11)**2
   40 Q(19)=Q(19)*Q(13)
      RETURN
      END
```

```
$IBFTC TG050X  NOREF
C     SUBROUTINE GVEC COMPUTES FORCING FUNCTION -- SEE APPENDIX B
      SUBROUTINE GVEC
      DIMENSION DS(3),D2(3)
      COMMON /DNBY/ RZERO(3),VZERO(3),PM(3),VM(3),Q(23),PS(3),G(3),K(6),
     1ACCREC,ACCUI,ACCLI,US,UE,UM,CJ2,DELM,R,V,DELS,DM(3),TMSTOP,
     2TIMEA,TIME,CHOVER,HT,HE,RPRES(3),VPRES(3),ESTOP,DELTE,AA(3,3,5),
     3IPRINT,AO(6,6),ICYCLE,IMOON,ISUN,TMOON(32,25),TSUN(4,16),DE(3)

C     DM= VEHICLE-MOON POSITION VECTOR
C     DE=VEHICLE-EARTH POSITION VECTOR

      CALL ROOT(PM(1),RMSQ,RM,1)
      RM3=RMSQ*RM
      IF(K(1)) 10,10,12
   10 DO 11 I=1,3
      DM(I)=RPRES(I)-PM(I)
   11 DE(I)=RPRES(I)
      GO TO 14
   12 DO 13 I=1,3
      DM(I)=RPRES(I)
   13 DE(I)=RPRES(I)+PM(I)
   14 DO 15 I=1,3
      DS(I)=DE(I)-PS(I)
   15 D2(I)=PM(I)-PS(I)
      CALL ROOT(DE(1),RSQ,R,1)
      CALL ROOT(DM(1),DELMSQ,DELM,1)
      CALL ROOT(DS(1),DELSQ,DELS,1)
      CALL ROOT(D2(1),DEL2SQ,DEL2,1)
      DELS3=DELSQ*DELS
      U3=-US/DELS3
      R3=RSQ*R
      IF(K(1)) 16,16,17
   16 DELM3=DELMSQ*DELM
      CALL ROOT(PS(1),RSSQ,RS,1)
      RS3=RSSQ*RS
      U1=-UM/DELM3
      U2=-UM/RM3
      U4=-US/RS3
      GO TO 18
   17 DEL23=DEL2SQ*DEL2
      U1=-UE/R3
      U2=UE/RM3
      U4=US/DEL23
   18 DO 19 I=1,3
   19 G(I)=U2*PM(I)+U3*DS(I)
      IF(K(1))20,20,22
   20 BRK=1.0-5.0*RPRES(3)**2/RSQ
      CO=-UE*CJ2/(R3*RSQ)
      DO 21 I=1,3
   21 G(I)=G(I)+DE(I)*BRK*CO  +U1*DM(I)+U4*PS(I)
      G(3)=G(3)+2.0*DE(3)*CO
      GO TO 24
   22 DO 23 I=1,3
   23 G(I)=G(I)+U1*DE(I)+U4*D2(I)
   24 R=R
      RETURN
      END
```

```
$IBFTC TG050Y  NOREF
C       SUBROUTINE PARTLS J D MCLEAN COMPUTES TWO BODY TRANSITION
C      MATRICES -- SEE APPENDIX A
      SUBROUTINE PARTLS
      DIMENSION QR(5),C(18),QQ(5),S(3),D2(4),D3(4),D4(4)
      EQUIVALENCE (D2(1),C(2)),(D3(1),C(8)),(D4(1),C(9))
      COMMON /DNBY/ RZERO(3),VZERO(3),PM(3),VM(3),Q(23),PS(3),G(3),M(6),
     1ACCREC,ACCUI,ACCLI,US,UE,UM,CJ2,DELM,R,V,DELS,DM(3),TMSTOP,
     2TIMEA,TIME,CHOVER,HT,HE,RPRES(3),VPRES(3),ESTOP,DELTE,AA(3,3,5),
     3IPRINT,AO(6,6),ICYCLE,IMOON,ISUN,TMOON(32,25),TSUN(4,16),DE(3)
      DO 90 I=1,5
      QR(I)=Q(I+3)
   90 QQ(I)=Q(I+4)
      C(17)=Q(10)/Q(12)
      C(16)=Q(9)/Q(16)
      C(15)=Q(12)/(Q(9)*Q(21))
      C(14)=1.0-Q(8)
      C(7)=-Q(7)/Q(15)
      C(9)=Q(17)*C(16)
      C(2)=Q(2)*C(17)
      C(5)=Q(4)/(Q(9)*Q(13)*Q(16))
      C(10)= 3.0*C(17)*Q(1)
      C(12)= Q(3)*C(17)
      C(18)=C(9)*(C(10)+Q(18)*C(12)+2.0*Q(19)*C(2))
      C(3)= Q(17)*(Q(3)*C(18)-2.0*C(2))
      C(1)=(C(3)-Q(22)*C(2))*C(15)+C(9)*Q(3)**2/Q(21)
      C(6)=Q(7)*C(17)+C(5)*C(18)
      C(10)=-Q(14)*(C(10)-Q(2)*C(18))
      C(4)=C(6)*C(15)-C(5)*Q(13)*Q(7)-Q(7)/Q(21)
      C(9)=-C(14)*C(2)*Q(22)
      C(2)= C(2)*Q(22)*Q(7)
      C(5)=C(5)*C(14)*Q(13)
      DO 9 I=1,3
    9 C(I+10)=C(I)*C(16)
      C(11)=C(11)-C(14)/Q(21)
      C(14)=C(14)/Q(15)
      C(8)=C(15)*C(10)+C(2)
      DO 12 K=1,5
      IF(K-3) 10,12,10
   10 DO11 I=1,3
      DO 51 J=1,3
   51 AA(I,J,K)=(C(K)*RZERO(I)+D3(K)*VZERO(I))*RZERO(J) +
     1 (D2(K)*RZERO(I)+D4(K)*VZERO(I))*VZERO(J)
      IF(K-3) 53,12,52
   52 AA(I,I,K)=AA(I,I,K)+QR(K)
      GO TO 11
   53 AA(I,I,K)=AA(I,I,K)+QQ(K)
   11 CONTINUE
   12 CONTINUE
      DO 41 I=1,3
   41 S(I)=C(7)*RZERO(I)+C(14)*VZERO(I)
      DO 13 I=1,3
      SR=0.0
      SV=0.0
      DO 14 J=1,3
      SR=SR+RPRES(J)*AA(J,I,1)
   14 SV=SV+RPRES(J)*AA(J,I,2)
      DO 13 K=1,3
      AA(K,I,4)=AA(K,I,4)+S(K)*SR
   13 AA(K,I,5)=AA(K,I,5)+S(K)*SV
      DO 17 I=1,3
      DO 17 J=1,3
      AO(I,J)=AA(I,J,1)
      AO(I+3,J+3)=AA(I,J,5)
      AO(I,J+3)= AA(I,J,2)
   17 AO(I+3,J)=AA(I,J,4)
   50 RETURN
      END
```

```
C     GENERAL PURPOSE SUBROUTINES


$IBFTC TG050B  NODECK,NOREF
C     MATRIX MULTIPLY ROUTINE
CTG050B SUBROUTINE XPYXM J D MCLEAN  J=1 C=A*B  J=2 C=A*BT  J=3 C=AT*B
      SUBROUTINE XPYXM(A,B,C,NRA,NCA,NRB,NCB,J)
C     NRA=NO. OF ROWS IN A
C     NCA= NO. OF COLUMNS IN A,   ETC


$IBFTC TG050D
CTG050D SUBROUTINE ROOT   J D  MCLEAN
      SUBROUTINE ROOT(X,RSQ,R,NFCA)
C     SUBROUTINE ROOT COMPUTES MAGNITUDE, R, AND SQUARE, RSQ, OF VECTOR
C     GIVEN BY ANY 3 CONSECUTIVE TERMS OF ARRAY, X, STARTING WITH
C     X(NFCA)


$IBFTC TG050S
C       LOADS COEFFICIENTS FOR MOON AND SUN POLYNOMIALS  L MCGEE
      SUBROUTINE MSLOAD(TMOON,LTABM,TSUN,LTABS,DATE,KM,KS)
C     THE CALLING PROGRAM WILL ORDINARILY HAVE TMOON DIMENSIONED AS
C     TMOON(LTABM,25) AND TSUN DIMENSIONED AS TSUN(LTABS,16) WHERE
C     LTABM AND LTABS ARE THE NUMBER OF ROWS IN THEIR RESPECTIVE TABLES.


$IBFTC TG050T
C        COMPUTES SUN POSITION FROM POLYNOMIAL COEFFICIENTS  L MCGEE
      SUBROUTINE SUNP(TIME,PS,TSUN,LTABS,NOSPTS,KERR,I)
C     PS IS POSITION VECTOR OF SUN, TIME IS FROM START OF TABLE IN SEC.


$IBFTC TG050U
C       COMPUTES MOON POSITION AND VELOCITY FROM POLYNOMIAL  L MCGEE
      SUBROUTINE MOON (TIME,QM,TMOON,LTABM,NOMPTS,KERR,POV,I)
C     QM IS POSITION OR VELOCITY VECTOR OF MOON DEPENDING ON WHETHER
C     POV IS POS OR VEL IN CALLING PROGRAM
```

REFERENCES

1. Danby, J. M. A.: Integration of the Equations of Planetary Motion in Rectangular Coordinates. Astron. J., vol. 67, no. 5, June 1962, pp. 287-299.

2. Danby, J. M. A.: Matrix Methods in the Calculation and Analysis of Orbits. AIAA J., vol. 2, no. 1, Jan. 1964, pp. 13-16.

3. Stone, C. R.; Johnson, Charles W.; Smith, Fred B.; Lee, E. Bruce; and Harvey, Charles A.: Time-Optimal Control of Linear Systems. Aero. Document R-ED6134, Minneapolis-Honeywell Regulator Co., Sept. 1, 1959.

4. Smith, Gerald L.; Schmidt, Stanley F.; and McGee, Leonard A.: Application of Statistical Filter Theory to the Optimal Estimation of Position and Velocity on Board a Circumlunar Vehicle. NASA TR R-135, 1962.

5. White, John S.: Simplified Calculation of Transition Matrices for Optimal Navigation. NASA TN D-3446, 1966.

6. Scarborough, James B.: Numerical Mathematical Analysis. Fourth ed., Johns Hopkins Press, 1958.

7. McLean, John D.; Schmidt, Stanley F.; and McGee, Leonard A.: Optimal Filtering and Linear Prediction Applied to a Midcourse Navigation System for the Circumlunar Mission. NASA TN D-1208, 1962.

8. McLean, J. D.: An Approximate Method for Computing Error Coefficient Matrices for Lunar Trajectories. Progress in Astronautics and Aeronautics, vol. 14, Academic Press, 1964.

9. Herget, P.: The Computation of Orbits. (Univ. of Cincinnati), Private Publications, 1948.

10. Goodyear, W. H.: Completely General Closed-Form Solution for Coordinates and Partial Derivatives of the Two-Body Problem. Astronom. J., vol. 70, no. 3, April 1965, pp. 189-192.

TABLE I.- TIME REQUIRED FOR ITERATIVE SOLUTION OF KEPLER'S EQUATION

Trajectory initial conditions	Precision of conic solution	Accuracy to one part in -	Time for 100 solutions, sec
Transearth	Single	10^7	3.2
Translunar	Single	10^7	2.0
Transearth	Double	10^8	8.0
Translunar	Double	10^8	4.4

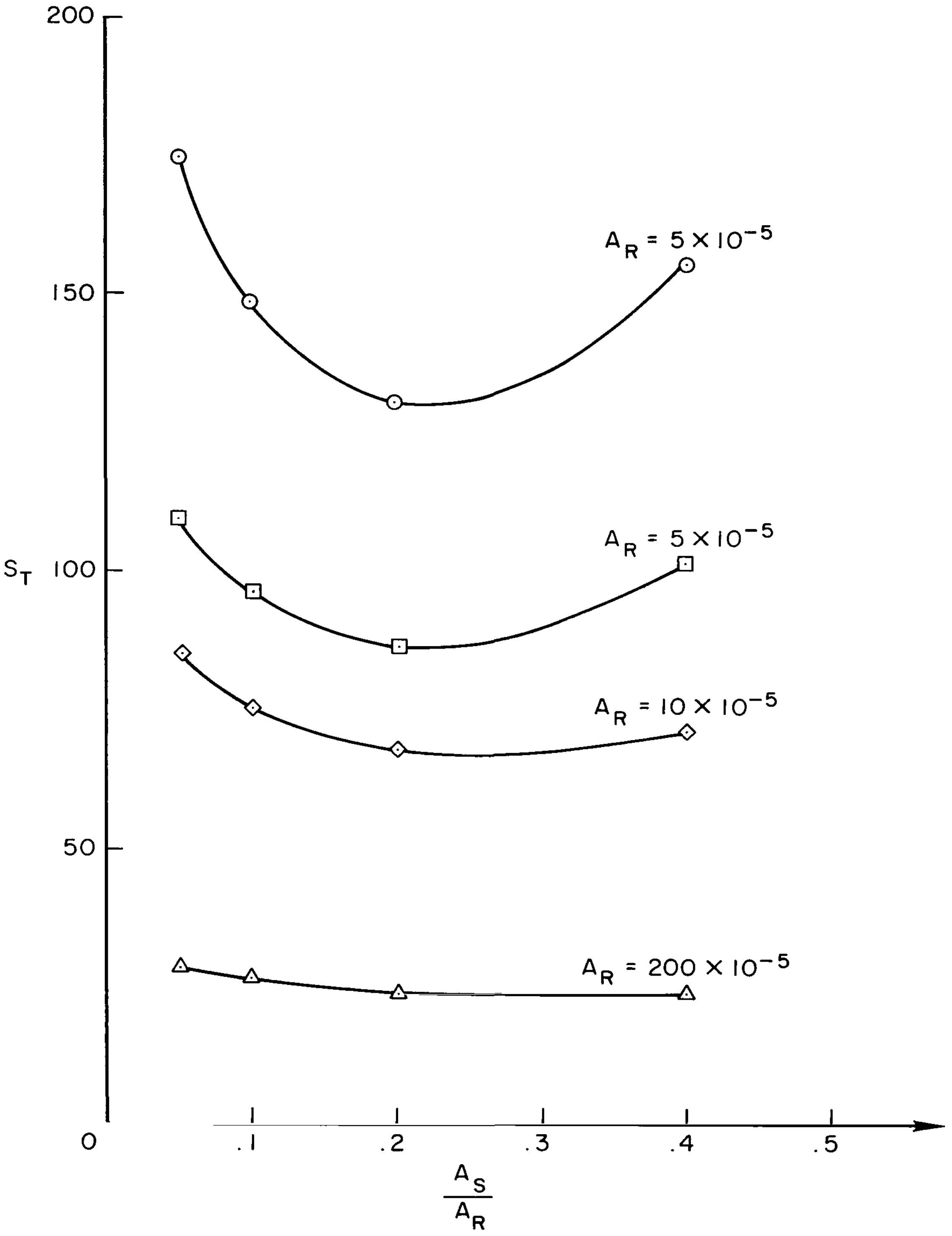


Figure 1.- Total number of integration steps with step size adjusted only by constraint on r/R.

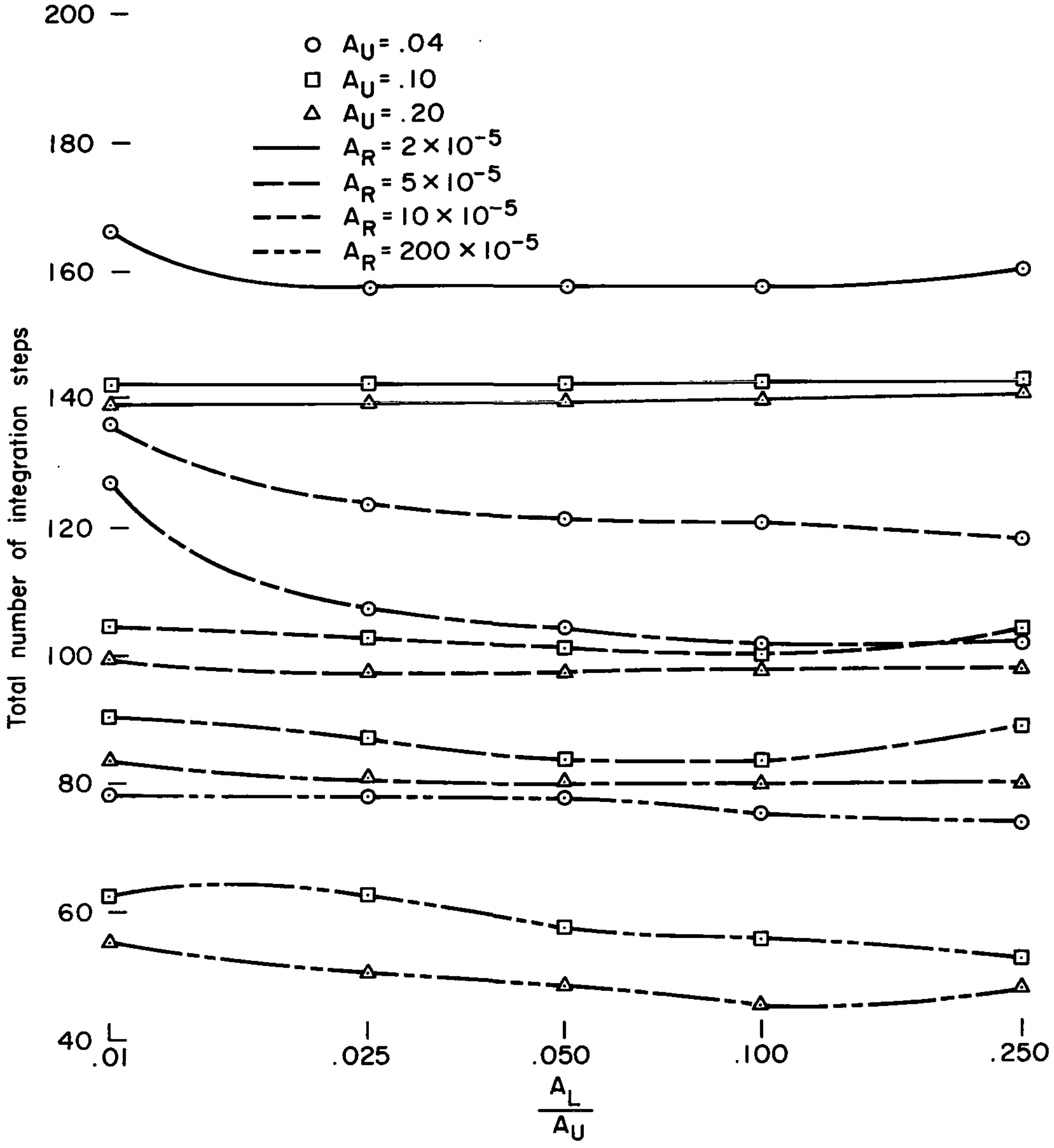


Figure 2.- Total number of integration steps with step size adjusted by constraints on r/R and A_Q.

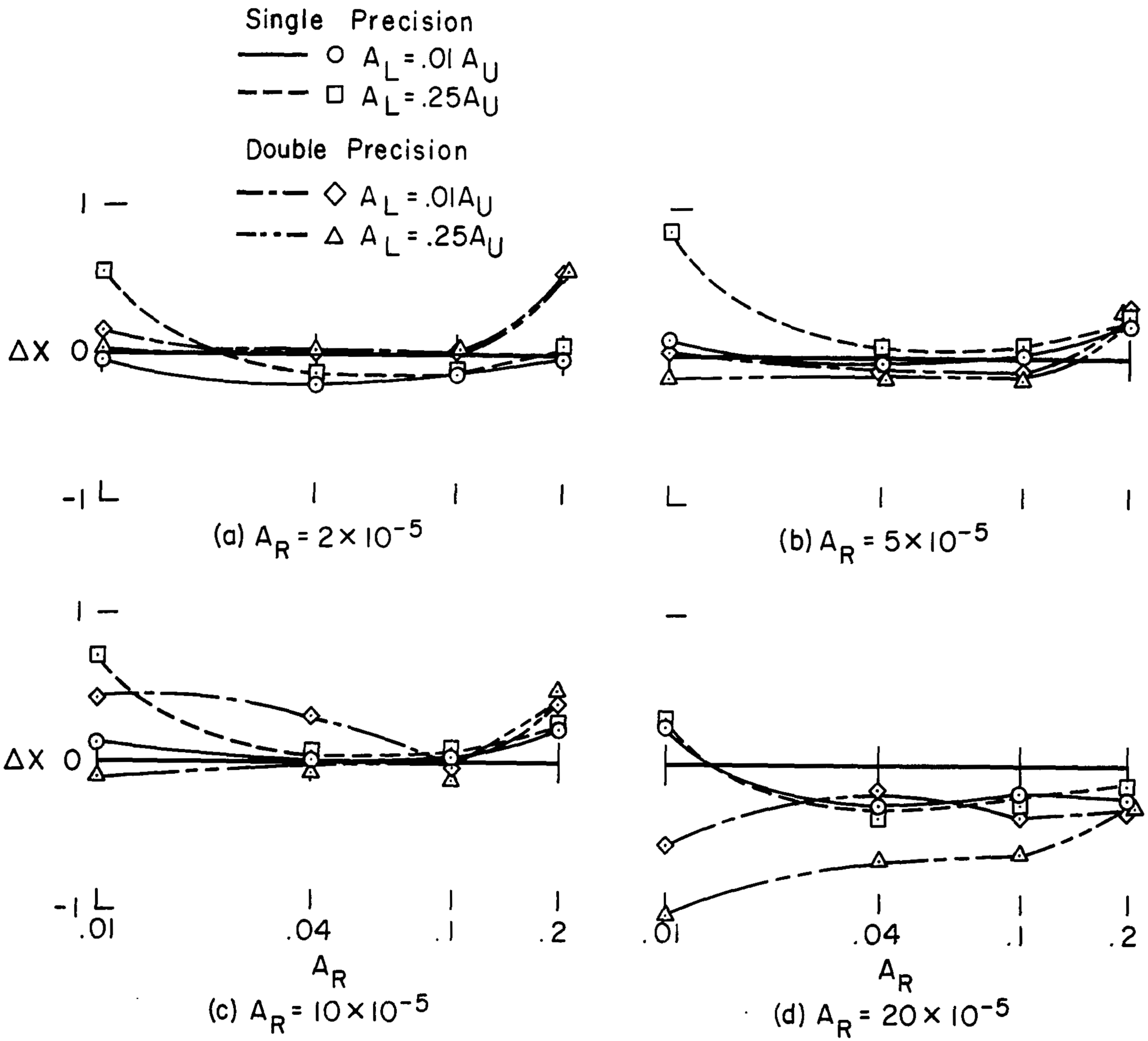


Figure 3.- Deviations in X component of position.

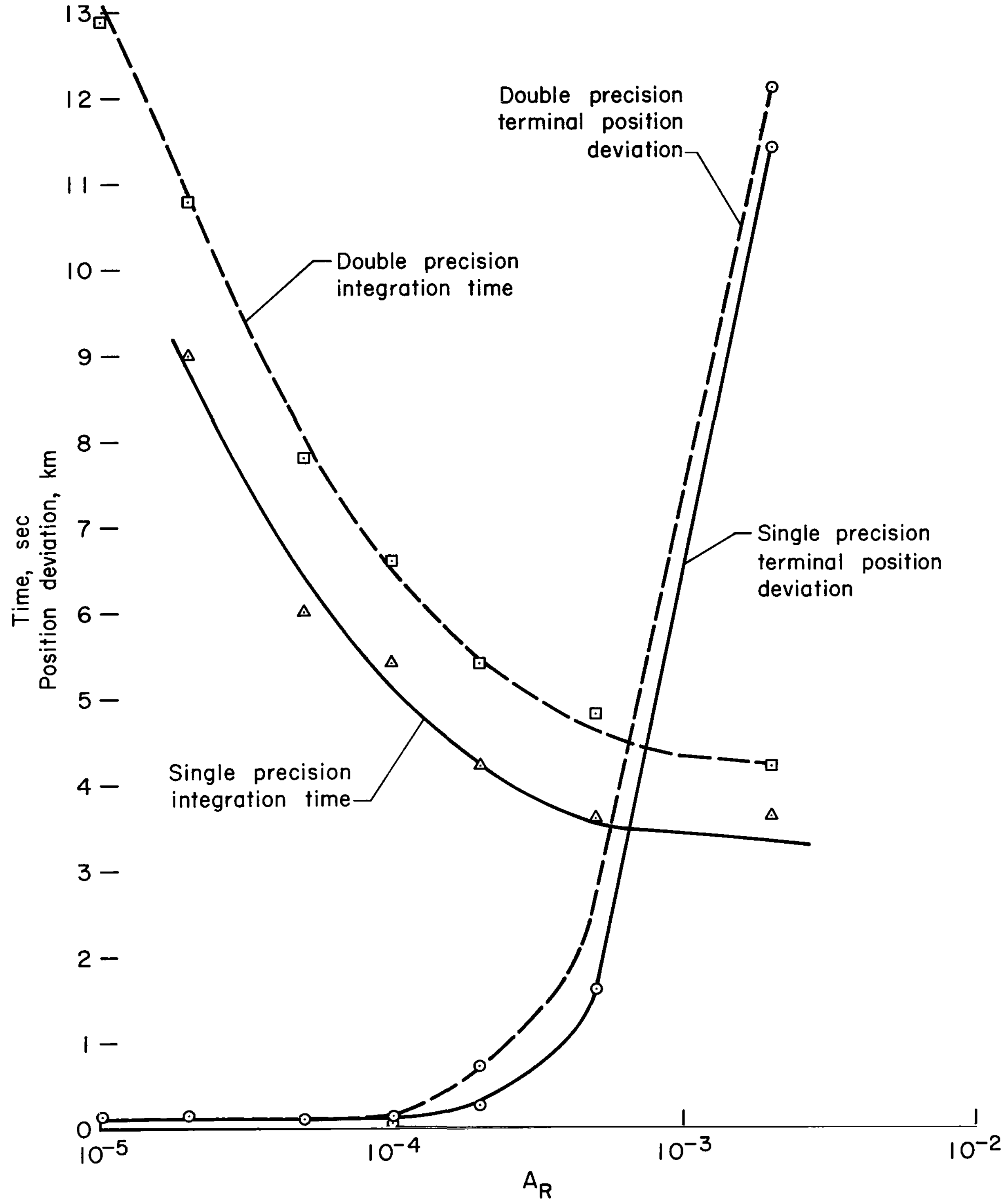


Figure 4.- Accuracy and integration time for transearth trajectory.

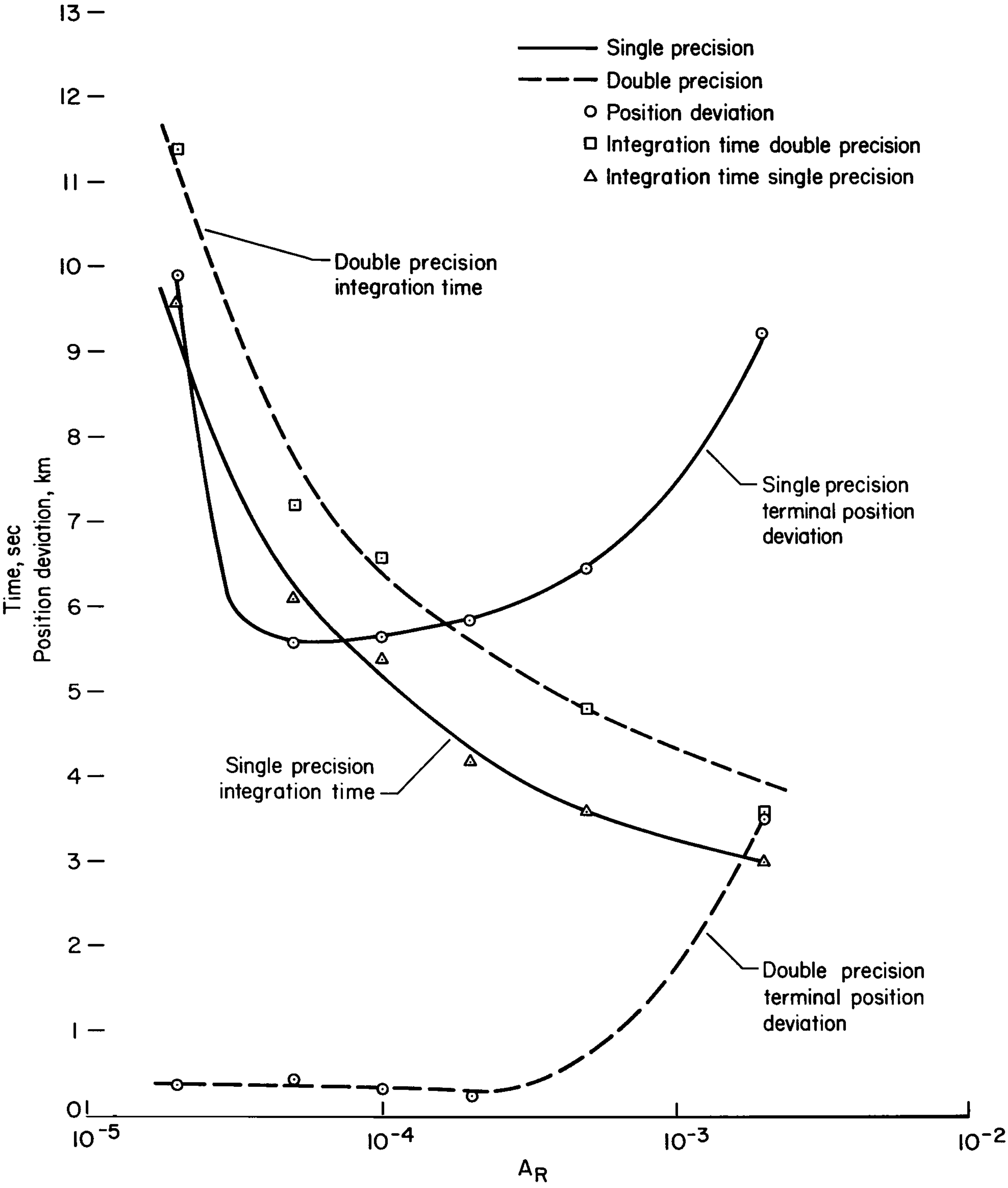


Figure 5.- Accuracy and integration time for translunar trajectory.

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Technical information generated in connection with a NASA contract or grant and released under NASA auspices.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

TECHNICAL REPRINTS: Information derived from NASA activities and initially published in the form of journal articles.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities but not necessarily reporting the results of individual NASA-programmed scientific efforts. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546